



中国科学技术大学
University of Science and Technology of China

BESIII介绍与Rhopi程序分析

参考裴宇鹏师兄的入门分析教程

王祎景

2023年7月6日



一、BESIII 探测器介绍

Beijing Electron Positron Collider II

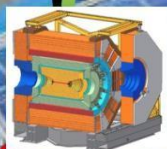


中国科学技术大学
University of Science and Technology of China

BEPCII / BESIII 鸟瞰图

BEPCII 储存环:
 e^+ 环和 e^- 环

直线加速器



BESIII 探测器

束流能量 1.0-2.45 GeV

同步辐射线站

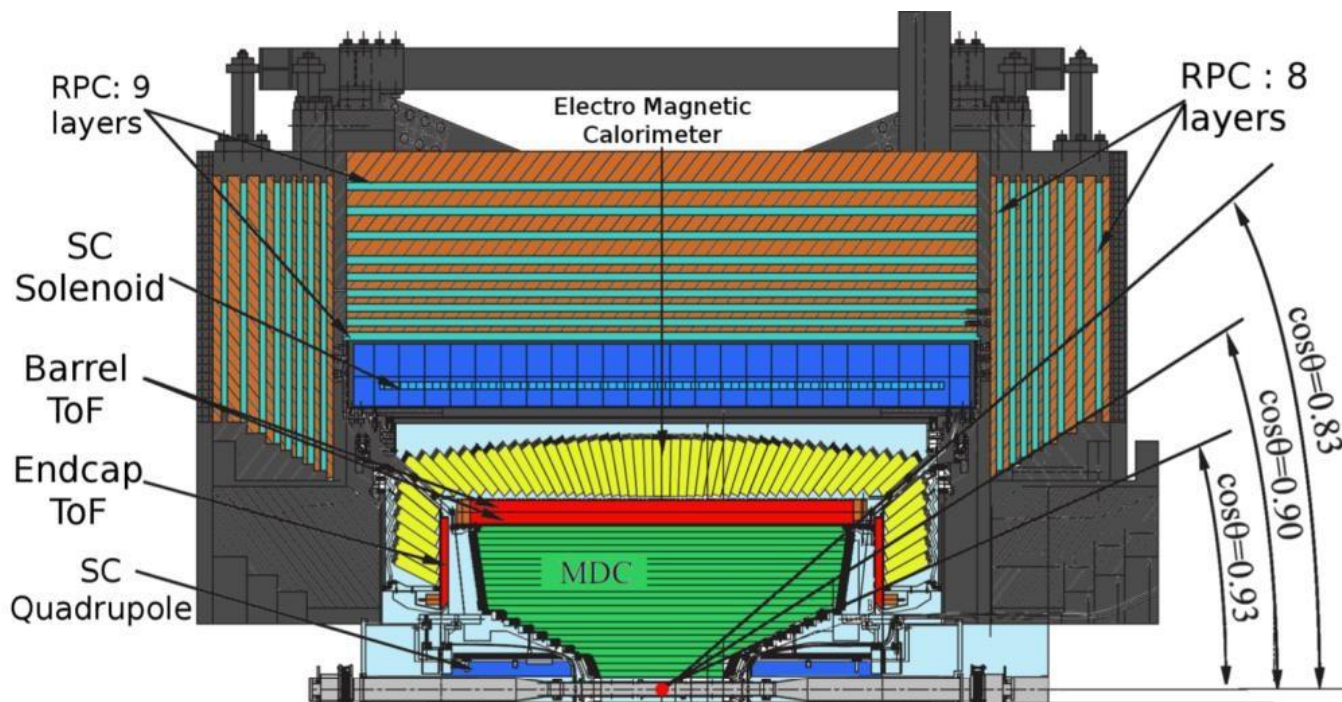
峰值亮度
 $1 \times 10^{33} / \text{cm}^2 / \text{s} @ \psi(3770)$

BESIII 探测器简介



中国科学技术大学
University of Science and Technology of China

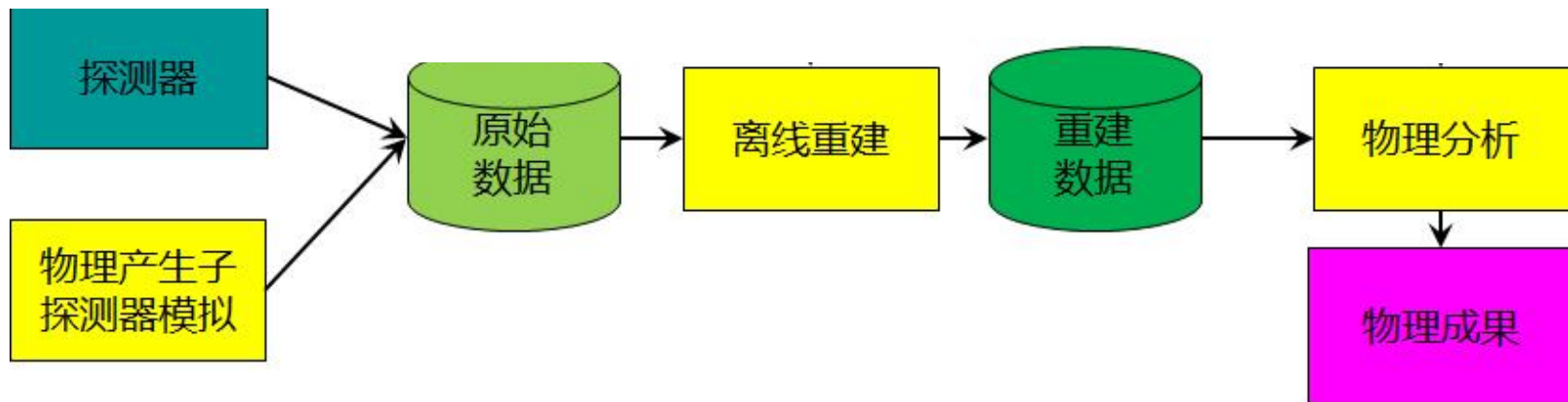
- **Main Draft Chamber (MDC)** : 主漂移室。测量带电粒子的径迹以及电离能损
- **Time of Flight (TOF)** : 飞行时间探测器。测量带电粒子的飞行时间
- **ElectroMagnetic Calorimeter (EMC)** : 电磁量能器。测量电磁型粒子能量。 $(e^{\pm}, \gamma)$
- **Super Conducting Solenoid** : 超导磁铁。提供1T磁场。
- **Muon Counter (MUC)** : 测量 μ 的位置，鉴别 μ/π 。



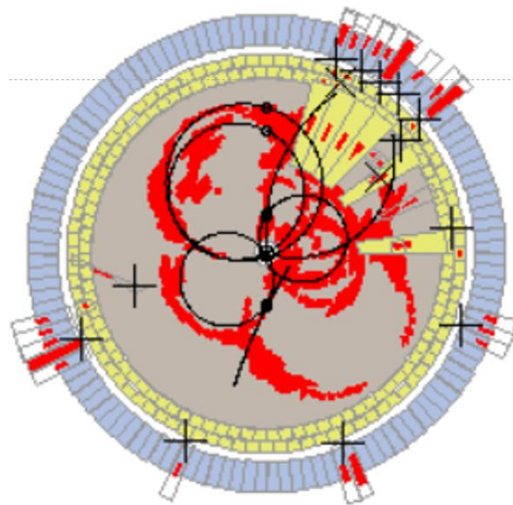
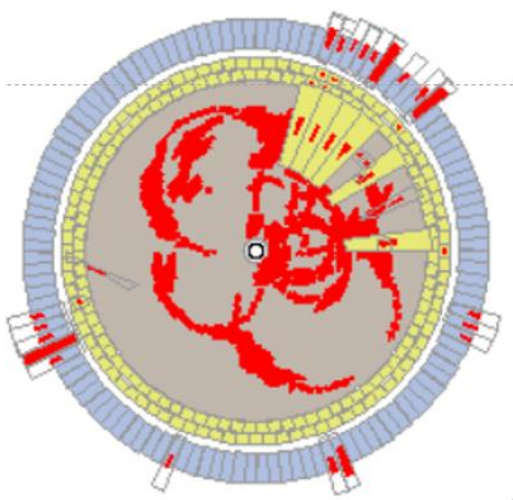
如何测量的呢？



- BESIII实验上可探测的末态粒子： $e, \mu, \pi, K, p, \gamma$



- 原始数据：**探测器给出的电子学信号（包含时间和幅度等信息）
- 重建数据：**通过重建算法，得到粒子的动量、能量等物理量



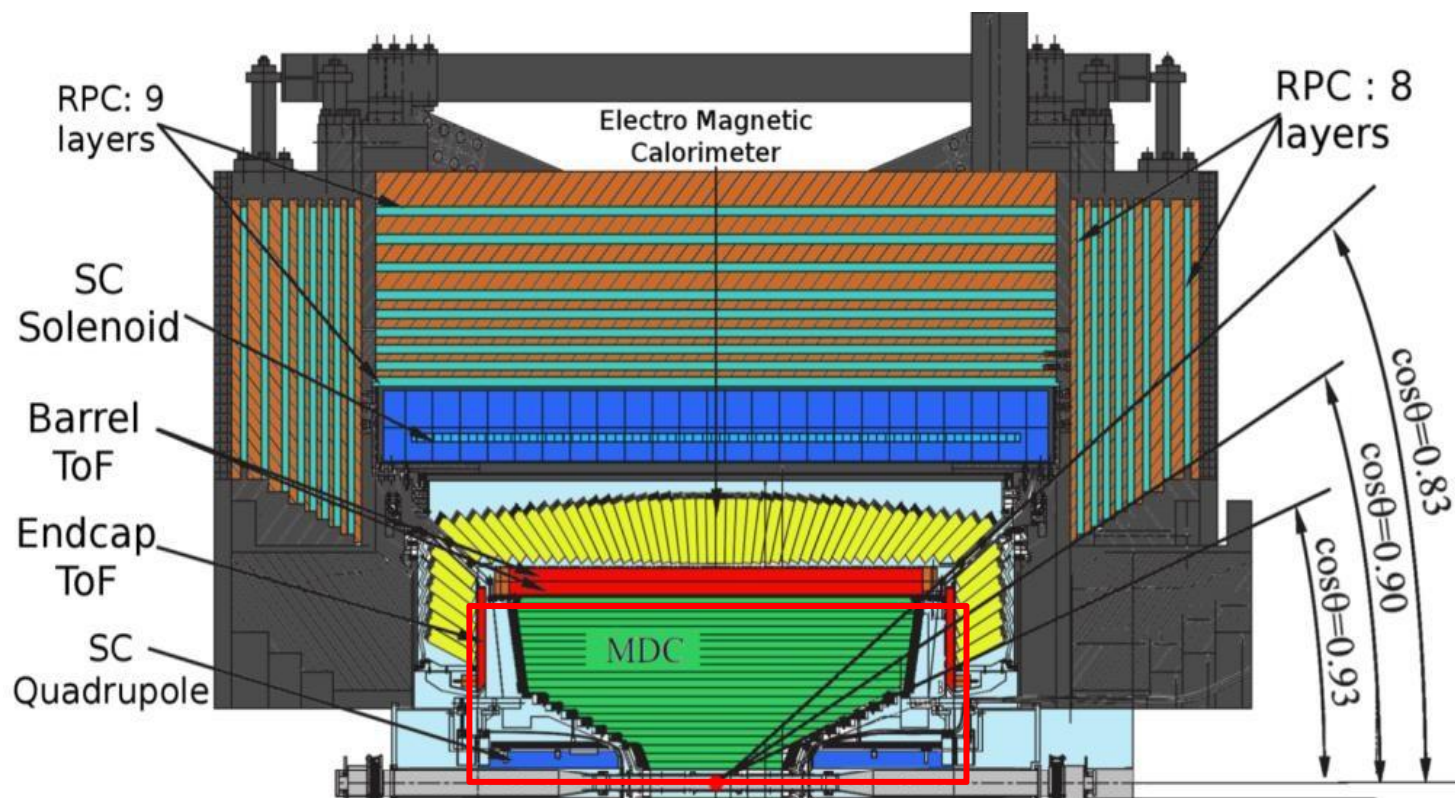
需要测量哪些物理量？



中国科学技术大学
University of Science and Technology of China

- **计数和计数率**，即粒子通量和流强
- **时间**，粒子到达探测器的时间或相对于某一时刻的时间间隔
- **能量**，粒子(特别是中性粒子)的能量
- **动量**，主要指带电粒子动量
- **位置**，粒子进入探测器的位置，带电粒子飞行径迹，中性粒子的位置与飞行方向
- **粒子的分辨**，通常使用 dE/dx 与TOF信息联合鉴别

- 带电粒子径迹的描述方法
- MDC工作原理
- 寻迹算法与径迹拟合



带电径迹五参数



中国科学技术大学
University of Science and Technology of China

- d_0 : 螺旋线于x-y平面投影与IP的距离
- ϕ_0 : x-y平面内, POCA的方位角
- κ : 横动量的倒数 ($1/P_t$), 其符号反映粒子电荷的正负
- z_0 : POCA的z坐标
- $\tan\lambda$: λ 为螺旋线与x-y平面的夹角
- IP (Initial Point): 正负电子对撞点
- POCA (Point of closest approach)

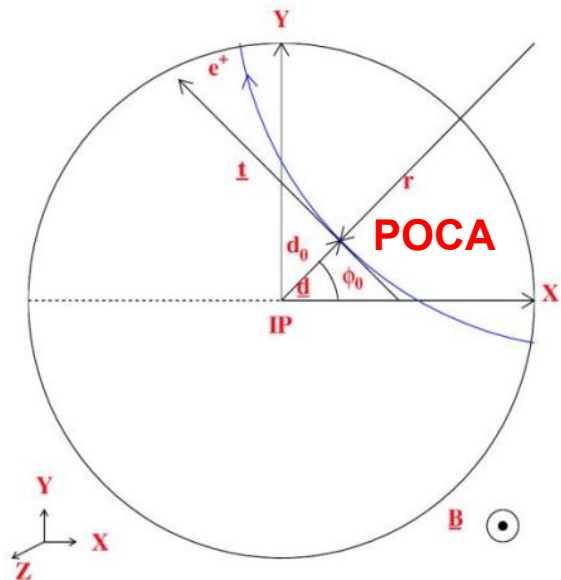
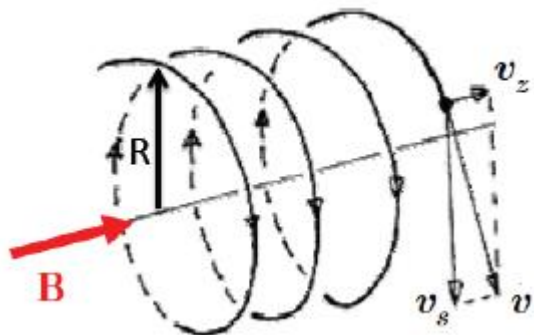
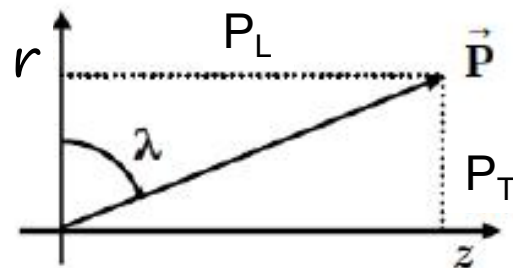


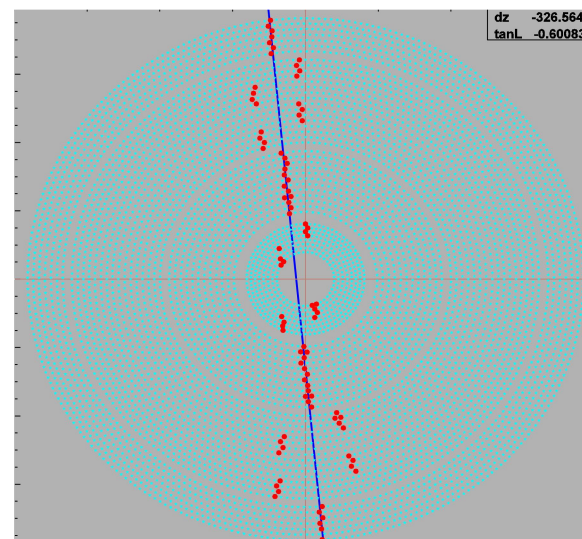
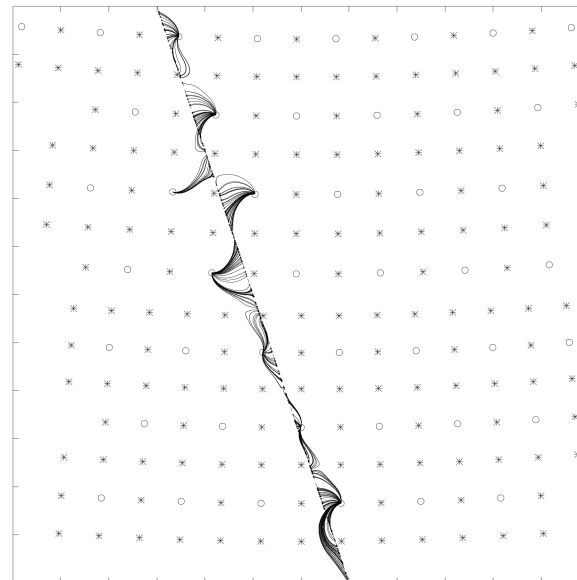
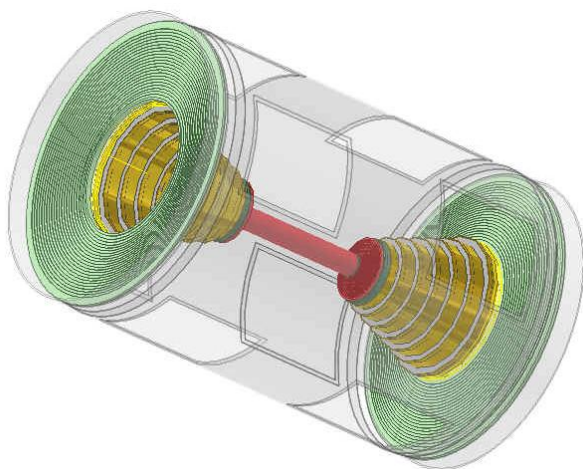
图 3.3 螺旋线径迹参数示意图



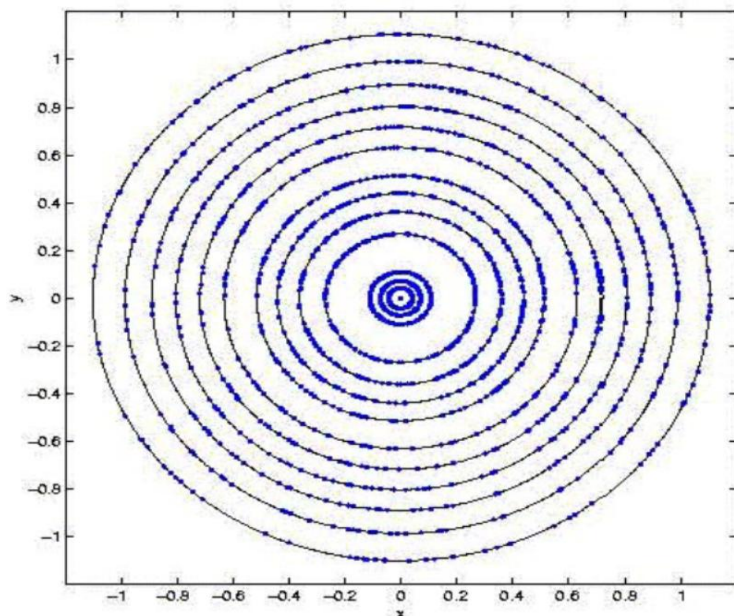
$$\mathbf{P} \equiv (d_0, \phi_0, \kappa, z_0, \tan\lambda)^T.$$



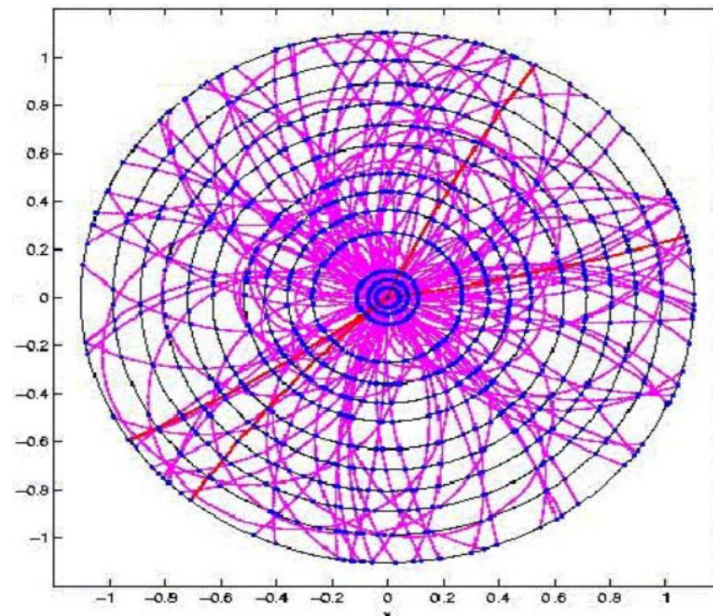
- 带电粒子穿过漂移室，通过电离激发产生电子—离子对，电子在电场下向阳极丝漂移感应信号。
- BESIII的漂移室拉有直丝与斜丝，分别定位径迹的横向与纵向。
- 通过寻迹算法将探测器上的击中转化为径迹信息。



探测器击中信息



重建的径迹



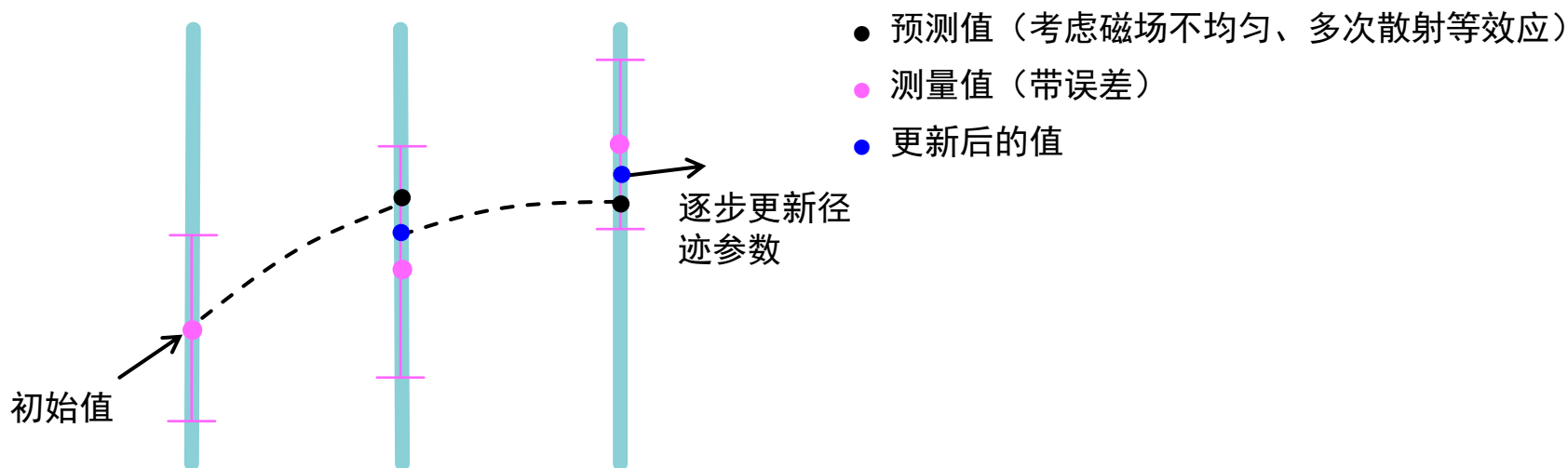
寻迹：将所有击中分类为一系列击中集合

- 局域寻迹算法
- 全局寻迹算法

拟合：得到径迹五参数

- 全局最小二乘法
- 卡曼滤波

- 递归的最小二乘估计，结合了寻迹与拟合
- 已知 → 预测 → 加入测量点 → 更新径迹 **拟合结果及误差矩阵**



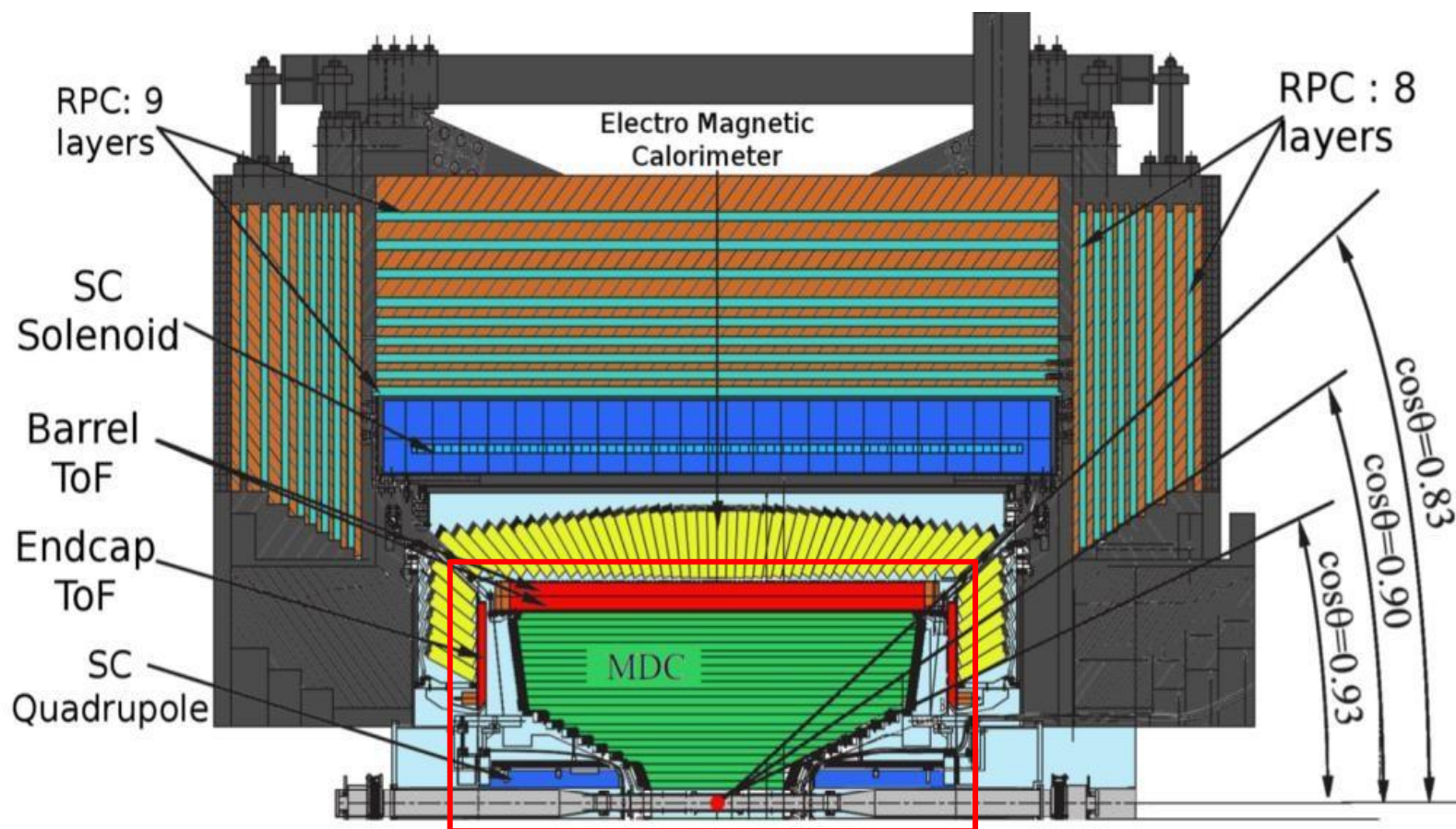
能量损失等与粒子种类相关，拟合结果与误差矩阵需在五种粒子假设下 (e, μ, π, K, p) 计算，物理分析中，得到正确的Kalman滤波径迹时，也要在正确的粒子假设下。

```
RecMdcKalTrack *pipTrk = (*(evtRecTrkCol->begin()+ipip[0]))->mdcKalTrack();
RecMdcKalTrack *pimTrk = (*(evtRecTrkCol->begin()+ipim[0]))->mdcKalTrack();

WTrackParameter wvpipTrk, wvpimTrk;
wvpipTrk = WTrackParameter(mpi, pipTrk->getZHelix(), pipTrk->getZError());
wvpimTrk = WTrackParameter(mpi, pimTrk->getZHelix(), pimTrk->getZError());
```


- MDC上的dEdx
- TOF上的飞行时间

(PID系统还包含EMC与MUC)



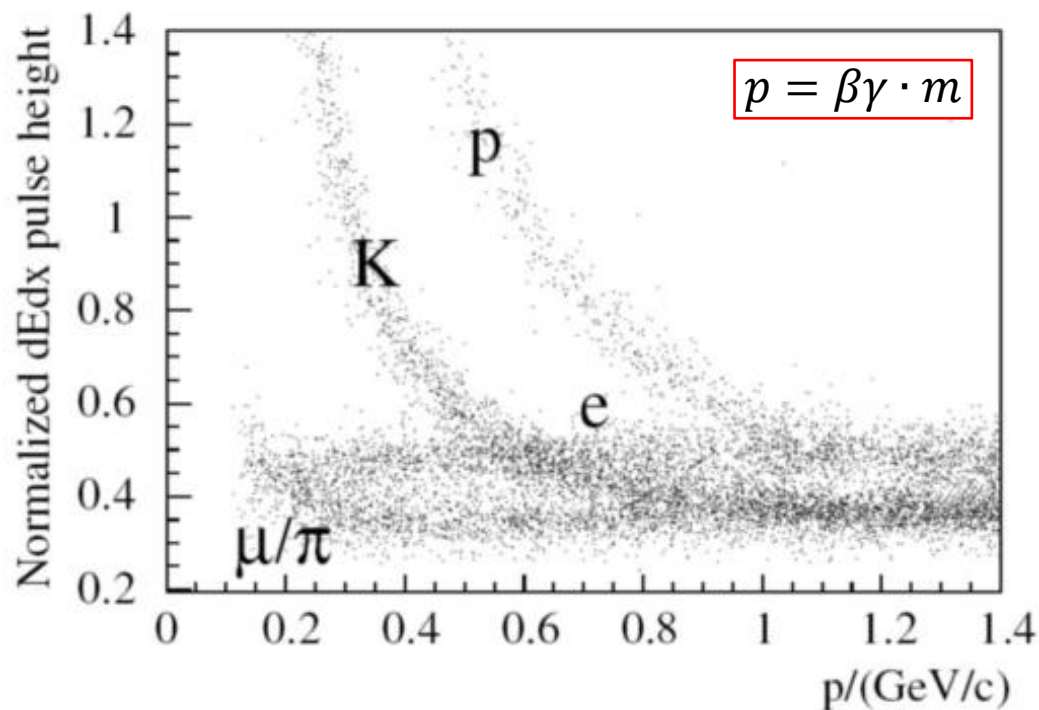
带电粒子与介质中的原子发生相互作用，损失的能量由Bethe-Bloch公式描述：

$$-\left\langle \frac{dE}{dx} \right\rangle = K z^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{\max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right]$$

低动量区： $-\frac{dE}{dx} \propto \frac{1}{\beta^2}$ ，由漂移室得到粒子动量，再根据dEdx信息可实现粒子鉴别。

$$\chi = \frac{dE/dx(\text{测量}) - dE/dx(\text{期望})}{\sigma(dE/dx)}$$

不同粒子假设得到不同的期望值，与测量值比较可以得出被测粒子是某种粒子的几率



飞行时间探测器(TOF)



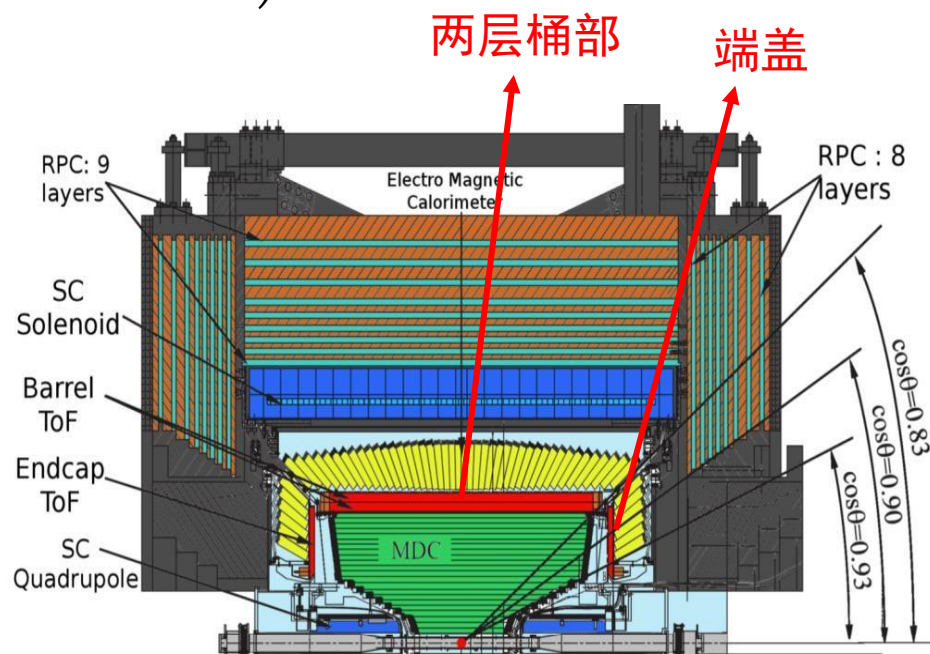
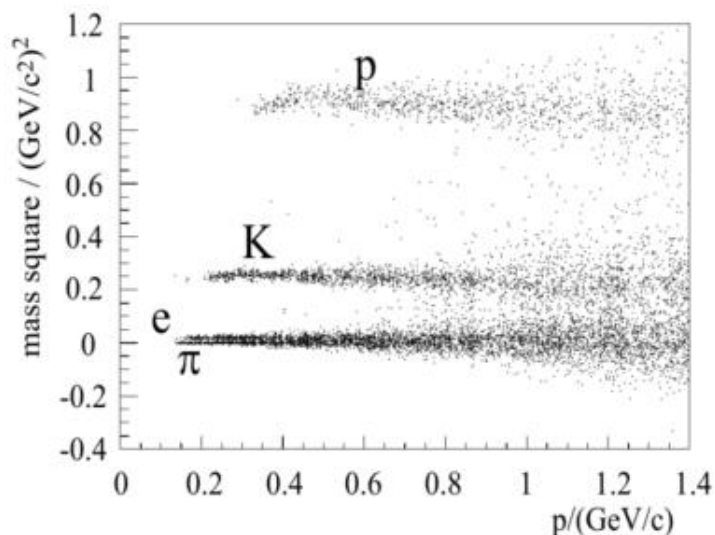
中国科学技术大学
University of Science and Technology of China

与dEdx类似，通过探测器测量得到的粒子动量 p 和飞行距离 L ，在粒子假设下，可计算得到预期飞行时间：

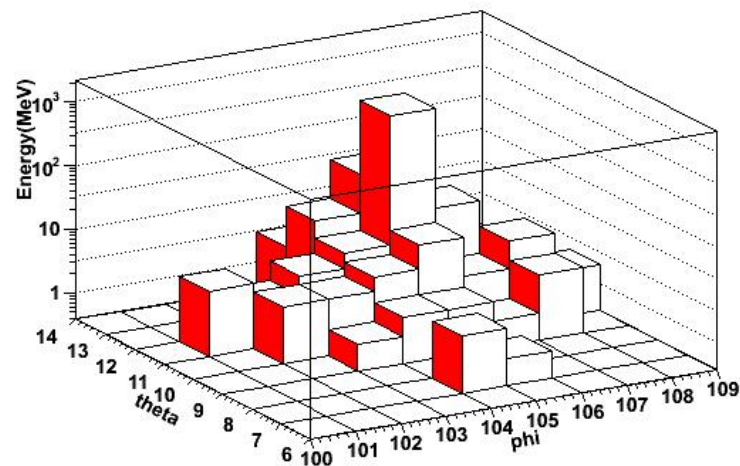
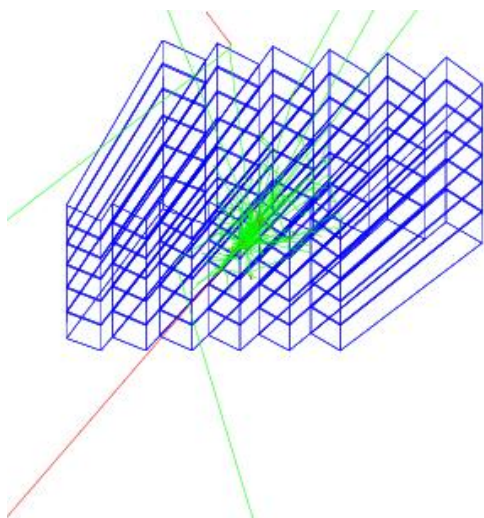
$$T_i^{Predict} = \frac{L}{c\beta_i}, \quad \beta_i = p/E_i, \quad E_i = \sqrt{m_i^2 + p^2}, \quad (i = e, \mu, \pi, K, p)$$

与测量时间比较：

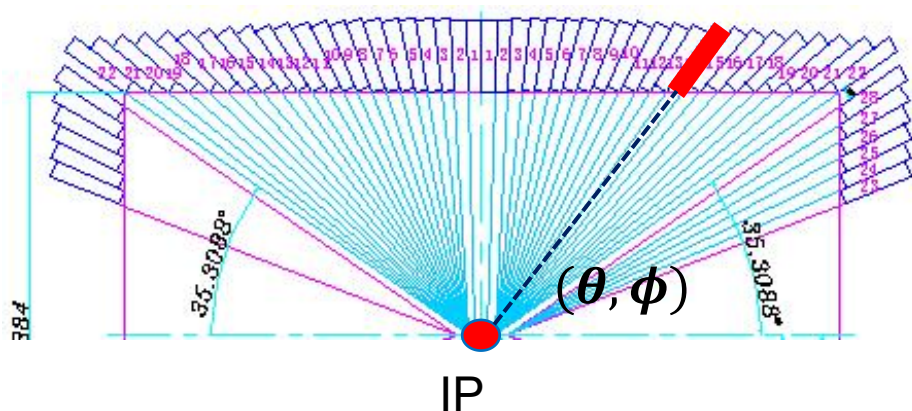
$$\chi^2 = \left(\frac{T^{Measure} - T_i^{Predict}}{\sigma} \right)^2$$



- 高能光子或正负电子于晶体中产生**电磁簇射**



- 光子四动量重建:

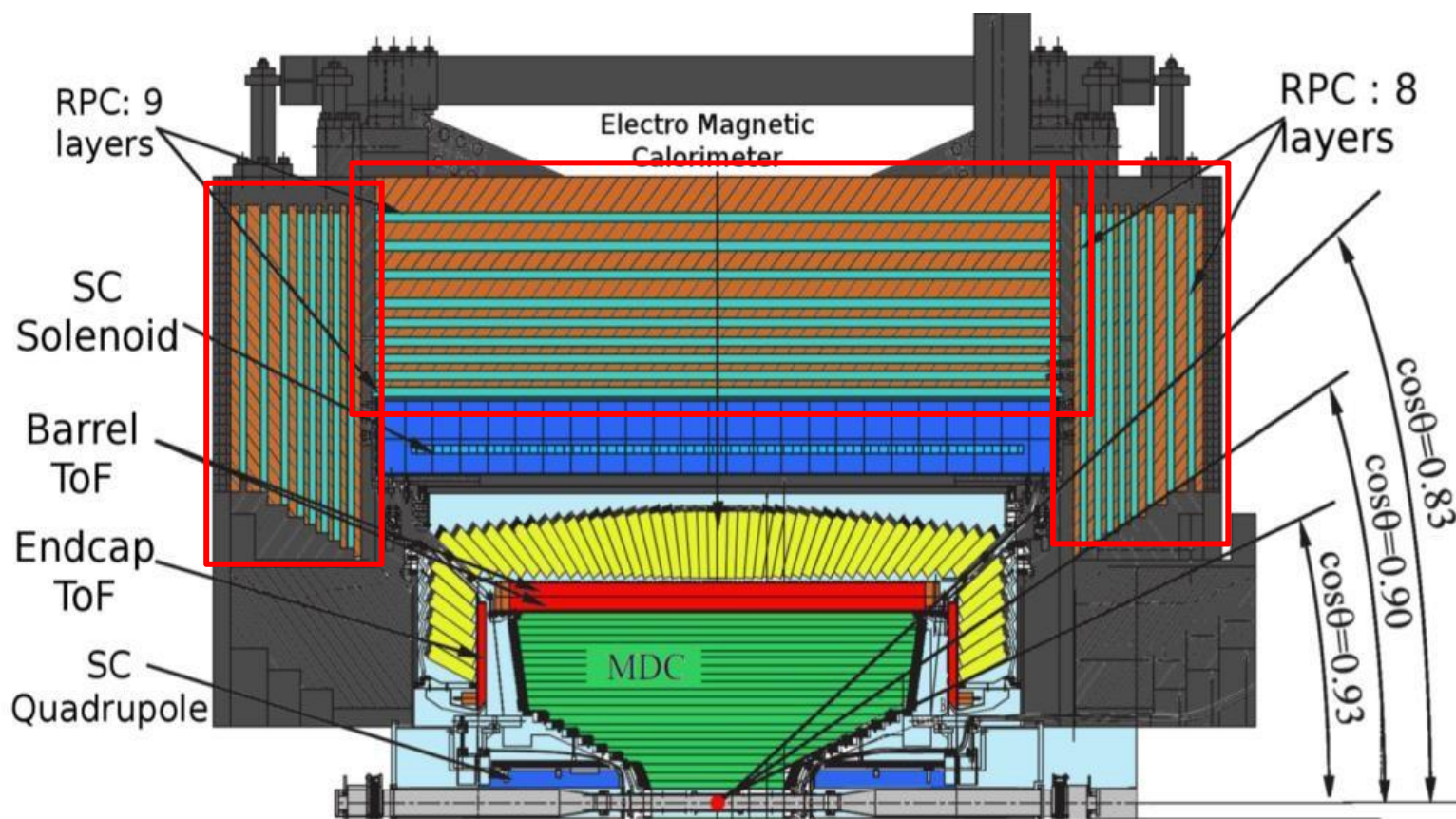


通过光子的位置信息得到其四动量:

$$P^\mu = (E, \vec{p})$$

$\vec{p}$ 由 (θ, ϕ) 确定。

- 测量末态 μ/π 的大概位置，与内层探测器的粒子径迹进行匹配
- 提高 μ/π 鉴别





- [Drift Chamber----Beijing Spectrometer\(BEIII\) Experiment \(cas.cn\)](http://cas.cn)
- <https://www.sciencedirect.com/science/article/pii/S0168900209023870>
- 《北京谱仪(BESIII)的设计与研制》王贻芳著



二、Rhopi分析算法介绍

分析算法包**不需要逐行理解**，一开始只需要理解每一块儿的内容，然后根据以后自己工作的需要再详细学习即可。

分析算法包用来做什么的???

重建事例

分析算法包



事例筛选

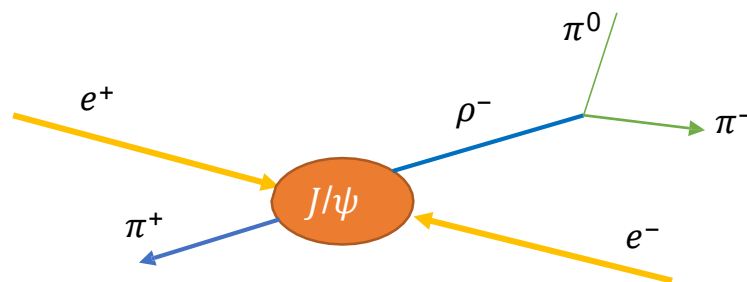
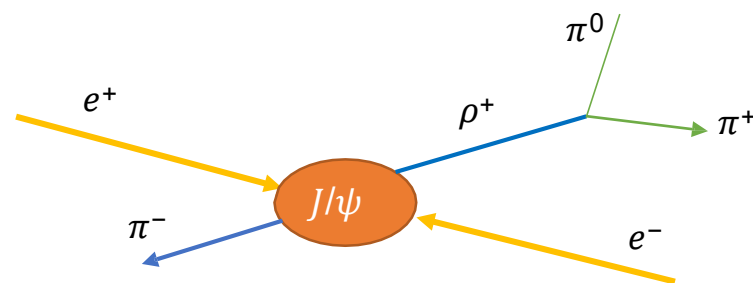
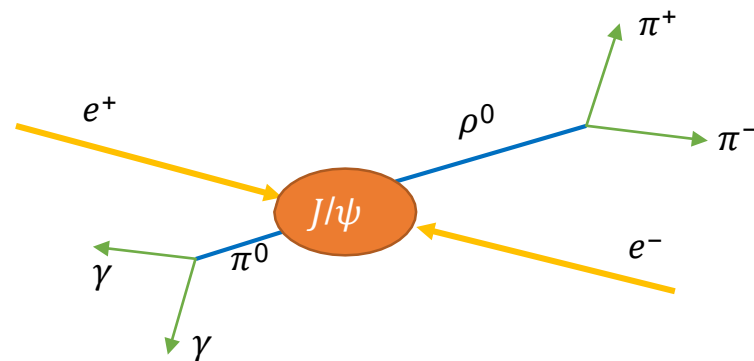
我们感兴趣的
物理事例

➤ 以rho pi算法包为例，级联衰变如下：

$$e^+e^- \rightarrow J/\psi$$

- $J/\psi \rightarrow \rho^0 \pi^0, \rho^0 \rightarrow \pi^+ \pi^-$
- $J/\psi \rightarrow \rho^+ \pi^-, \rho^+ \rightarrow \pi^+ \pi^0$
- $J/\psi \rightarrow \rho^- \pi^+, \rho^- \rightarrow \pi^- \pi^0$
- $\pi^0 \rightarrow \gamma \gamma$

末态粒子： $\pi^+ \pi^- \gamma \gamma$ (2个带电径迹和两个光子)





/src	源文件(*.cxx) (就是你的分析程序)
/ <package-name>	头文件(*.h) (里面包含程序中使用到的变量)
/cmt	编译的文件(cmt clean→cmt config→cmt make)
/share	jobOptions文件
/x86_bulabula	编译算法包后产生的

```
[tma@ui01 SigmaAlg-00-00-16]$ ll
total 4
drwxr-xr-x 2 tma BESIII 140 Nov 29 20:57 cmt
drwxr-xr-x 2 tma BESIII 42 Feb 7 21:14 share
drwxr-xr-x 2 tma BESIII 29 Feb 9 10:22 SigmaAlg
drwxr-xr-x 3 tma BESIII 66 Mar 2 14:19 src
drwxr-xr-x 2 tma BESIII 4096 Feb 9 10:44 x86_64-slc6-gcc46-opt
```

算法包路径:

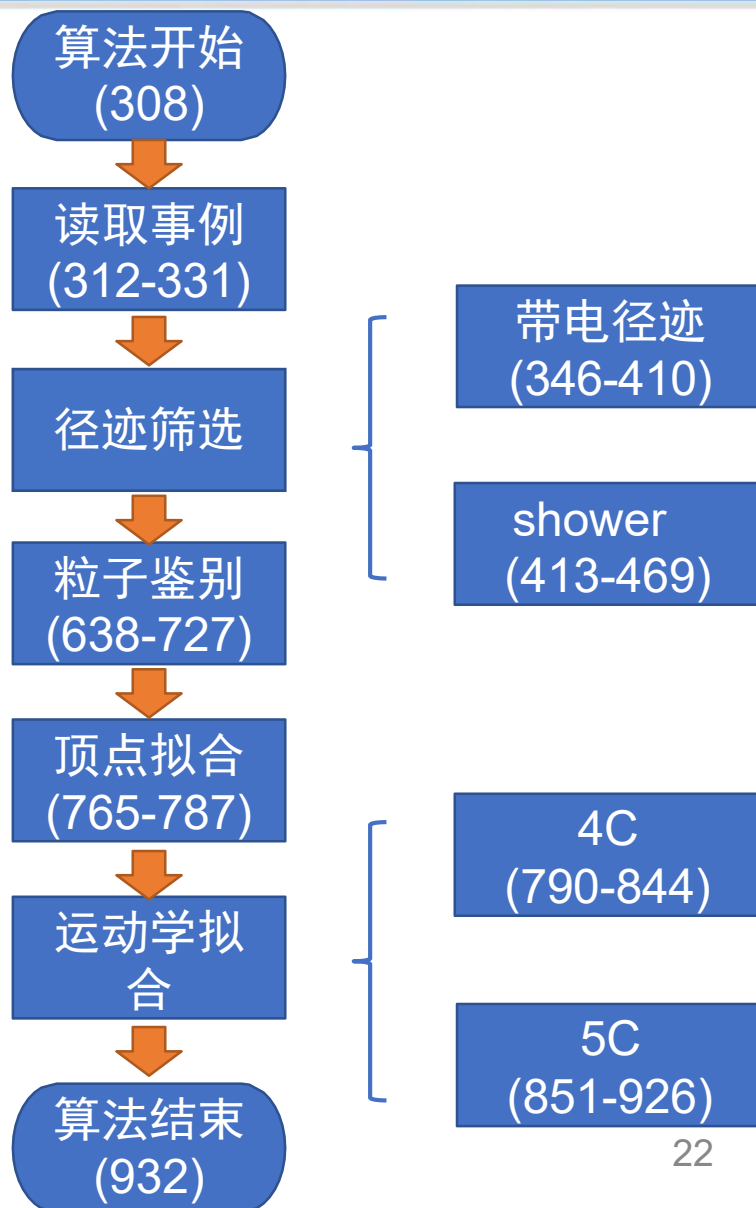
/cvmfs/bes3.ihep.ac.cn/bes3sw/Boss/7.1.0/Analysis/Physics/RhopiAlg

程序框架(*.CXX)



中国科学技术大学
University of Science and Technology of China

- initialize(): 初始化
- **execute():** 算法开始
- finalize(): 算法结束



- “开关”可以从ana.txt直接更改
- 避免多次编译
- 分析程序中设置default值，外部修改则以外部为主。

src/rhopi.cxx

```
59 Rhopi::Rhopi(const std::string& name, ISvcLocator* pSvcLocator) :  
60     Algorithm(name, pSvcLocator) {  
61  
62     //Declare the properties  
63     declareProperty("vr0cut", m_vr0cut=1.0);  
64     declareProperty("vz0cut", m_vz0cut=5.0);  
65     declareProperty("EnergyThreshold", m_energyThreshold=0.04);  
66     declareProperty("GammaPhiCut", m_gammaPhiCut=20.0);  
67     declareProperty("GammaThetaCut", m_gammaThetaCut=20.0);  
68     declareProperty("GammaAngleCut", m_gammaAngleCut=20.0);  
69     declareProperty("Test4C", m_test4C = 1);  
70     declareProperty("Test5C", m_test5C = 1);  
71     declareProperty("checkDedx", m_checkDedx = 1);  
72     declareProperty("checkTof", m_checkTof = 1);  
73 }
```

ana.txt (用于运行分析程序的脚本)

```
1 #include "$ROOTIROOT/share/joboptions_ReadRec.txt"  
2 #include "$VERTEXFITROOT/share/joboptions_VertexDbSvc.txt"  
3 #include "$MAGNETICFIELDROOT/share/MagneticField.txt"  
4 #include "$ABSCORROOT/share/joboptions_AbsCor.txt"  
5 #include "$MCDECAYMODESVCROOT/share/GetDecayMode.txt"  
6 #include "$MEASUREDECMSSVCROOT/share/anaOptions.txt"  
7 #include "/workfs2/bes/peiy/workarea/7.0.5/Analysis/Physics/RhopiAlg/RhopiAlg-00-00-23/share/joboptions_Rhopi.txt"  
8  
9 Rhopi.GammaAngleCut = 15;  
10 Rhopi.EnergyThreshold = 0.025;  
11 Rhopi.Test4C=1;  
12  
13 // Input REC or DST file name  
14 EventCnvSvc.digiRootInputFile =  
15  
16 "/besfs4/offline/data/705-1/jpsi/round11/mc/tag/tag1/jpsi2018_stream001_run52962_file1.dst"  
17  
18 };  
19  
20 // Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING, 5=ERROR, 6=FATAL )  
21 MessageSvc.OutputLevel = 5;  
22  
23 // Number of events to be processed (default is 10)  
24 ApplicationMgr.EvtMax = 10000;  
25  
26 ApplicationMgr.HistogramPersistency = "ROOT";  
27 NTuplesSvc.Output = { "FILE1 DATAFILE='res1.root' OPT='NEW'TYP='ROOT'"};
```


Initialize()



```
76 StatusCode Rhopi::initialize(){
77     MsgStream log(msgSvc(), name());
78
79     log << MSG::INFO << "in initialize()" << endl;
80
81     StatusCode status;
82     NTuplePtr ntl(ntupleSvc(), "FILE1/vxyz");
83     if ( ntl ) m_tuple1 = ntl;
84     else {
85         m_tuple1 = ntupleSvc()->book ("FILE1/vxyz", CLID_ColumnwiseTuple, "ks N-Tuple example");
86         if ( m_tuple1 ) {
87             status = m_tuple1->addItem ("vx0", m_vx0);
88             status = m_tuple1->addItem ("vy0", m_vy0);
89             status = m_tuple1->addItem ("vz0", m_vz0);
90             status = m_tuple1->addItem ("vr0", m_vr0);
91             status = m_tuple1->addItem ("rvxy0", m_rvxy0);
92             status = m_tuple1->addItem ("rvz0", m_rvz0);
93             status = m_tuple1->addItem ("rvphi0", m_rvphi0);
94         }
95         else {
96             log << MSG::ERROR << " Cannot book N-tuple:" << log(m_tuple1) << endl;
97             return StatusCode::FAILURE;
98         }
99     }
```

rhopi.cxx

vxyz即是root
中tree的名字

```
68 NTuple::Tuple* m_tuple1;
69 NTuple::Item<double> m_vx0;
70 NTuple::Item<double> m_vy0;
71 NTuple::Item<double> m_vz0;
72 NTuple::Item<double> m_vr0;
73 NTuple::Item<double> m_rvxy0;
74 NTuple::Item<double> m_rvz0;
75 NTuple::Item<double> m_rvphi0;
```

RhopiAlg/rhopi.h

```
[peiyp@xlslc708 test]$ r1 rhopi.root
root [0]
Attaching file rhopi.root as _file0...
root [1] .ls
TFile**      rhopi.root      Gaudi Trees
TFile*      rhopi.root      Gaudi Trees
KEY: TTree   vxyz;1         ks N-Tuple example
KEY: TTree   photon;1       ks N-Tuple example
KEY: TTree   dedx;1         ks N-Tuple example
KEY: TTree   tof1;1         ks N-Tuple example
KEY: TTree   pid;1          ks N-Tuple example
KEY: TTree   etot;1         ks N-Tuple example
KEY: TTree   tof2;1         ks N-Tuple example
KEY: TTree   fit4c;1        ks N-Tuple example
KEY: TTree   fit5c;1        ks N-Tuple example
KEY: TTree   geff;1         ks N-Tuple example
KEY: TTree   tofe;1         ks N-Tuple example
```

```
root [3] vxyz->Print()
*****
*tree :vxyz : ks N-Tuple example *
*Entries : 17277 : Total = 973718 bytes File Size = 936003 *
* : : Tree compression factor = 1.04 *
*****
*Br 0 :vx0 : vx0/D *
*Entries : 17277 : Total Size= 139040 bytes File Size = 133683 *
*Baskets : 5 : Basket Size= 32000 bytes Compression= 1.04 *
*
*Br 1 :vy0 : vy0/D *
*Entries : 17277 : Total Size= 139040 bytes File Size = 133853 *
*Baskets : 5 : Basket Size= 32000 bytes Compression= 1.04 *
*
*Br 2 :vz0 : vz0/D *
*Entries : 17277 : Total Size= 139040 bytes File Size = 134413 *
*Baskets : 5 : Basket Size= 32000 bytes Compression= 1.03 *
*
*Br 3 :vr0 : vr0/D *
*Entries : 17277 : Total Size= 139040 bytes File Size = 134160 *
*Baskets : 5 : Basket Size= 32000 bytes Compression= 1.03 *
*
*Br 4 :rvxy0 : rvxy0/D *
*Entries : 17277 : Total Size= 139058 bytes File Size = 132724 *
*Baskets : 5 : Basket Size= 32000 bytes Compression= 1.04 *
*
*Br 5 :rvz0 : rvz0/D *
*Entries : 17277 : Total Size= 139049 bytes File Size = 134401 *
*Baskets : 5 : Basket Size= 32000 bytes Compression= 1.03 *
*
*Br 6 :rvphi0 : rvphi0/D *
*Entries : 17277 : Total Size= 139067 bytes File Size = 131752 *
*Baskets : 5 : Basket Size= 32000 bytes Compression= 1.05 *
*****
```

- “m_”开头的变量需要在.h中声明，是程序中变量的名字
- 红色引号中的变量是外部root中的名字

execute()

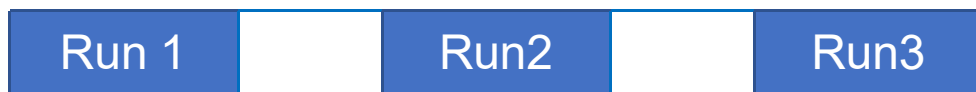


```

309 StatusCode Rhopi::execute() {
310
311     std::cout << "execute()" << std::endl;
312
313     MsgStream log(msgSvc(), name());
314     log << MSG::INFO << "in execute()" << endreq;
315
316     SmartDataPtr<Event::EventHeader> eventHeader(eventSvc(), "/Event/EventHeader");
317     int runNo=eventHeader->runNumber();
318     int event=eventHeader->eventNumber();
319     log << MSG::DEBUG << "run, evtNum = "
320         << runNo << ", "
321         << event <<endreq;
322     cout<<"event " <<event<<endl;
323     Ncut0++;
324
325     SmartDataPtr<EvtRecEvent> evtRecEvent(eventSvc(), EventMode1::EvtRec::EvtRecEvent);
326     // log << MSG::INFO << "get event tag OK" << endreq;
327     log << MSG::DEBUG << "ncharg, nneu, tottks = "
328         << evtRecEvent->totalCharged() << ", "
329         << evtRecEvent->totalNeutral() << ", "
330         << evtRecEvent->totalTracks() <<endreq;

```

- 每个事例有唯一的run号与event号
- 分析程序每次execute()仅处理一个event，处理完毕进入下一个event
- Event by Event !!!



	Track
0	π^+
1	π^-
2	γ
3	γ

RUN STATUS LIST

ID	run_number	start_time	end_time	total_event	status
51705	71150	2022-01-19 08:33:45	2022-01-19 09:35:02	5058422	Good
51704	71149	2022-01-19 07:23:56	2022-01-19 08:25:22	4851754	Good
51703	71148	2022-01-19 06:26:00	2022-01-19 07:15:42	4192731	Good
51702	71147	2022-01-19 05:03:42	2022-01-19 06:05:47	4925395	Good
51701	71146	2022-01-19 03:54:26	2022-01-19 04:55:44	4876583	Good
51700	71145	2022-01-19 02:43:56	2022-01-19 03:45:46	4976439	Good
51699	71144	2022-01-19 01:34:22	2022-01-19 02:35:42	4914805	Good
51698	71143	2022-01-19 00:24:50	2022-01-19 01:25:51	4905122	Good
51697	71142	2022-01-18 23:13:41	2022-01-19 00:13:55	4898783	Good
51696	71141	2022-01-18 22:03:33	2022-01-18 23:05:54	5020576	Good

Run Number : 71150 Status : Good

Start Time : 2022-01-19 08:33:45 End Time : 2022-01-19 09:35:02

Total number of Events : 5058422 initialied beam energy : 1.8932GeV

Magnet Current : 3369.5872A Lum(Begin Run) : 19646.742 Lum(End Run) : 19648.674

Run Task : psi(3770) beam type : collision

HIT_MAP : Normal

HV : Normal

Other Problems : NO

Other comment :

Shift Chief : Zhefei Tian 2022-01-19 09:38:30

```
362 for(int i = 0; i < evtRecEvent->totalCharged(); i++){
363     EvtRecTrackIterator itTrk=evtRecTrkCol->begin() + i;
364     if(!(*itTrk)->isMdcTrackValid()) continue;
365     RecMdcTrack *mdcTrk = (*itTrk)->mdcTrack();
366     double pch=mdcTrk->p();
367     double x0=mdcTrk->x();
368     double y0=mdcTrk->y();
369     double z0=mdcTrk->z();
370     double phi0=mdcTrk->helix(1);
371     double xv=xorigin.x();
372     double yv=xorigin.y();
373     double Rxy=(x0-xv)*cos(phi0)+(y0-yv)*sin(phi0);
374     m_vx0 = x0;
375     m_vy0 = y0;
376     m_vz0 = z0;
377     m_vr0 = Rxy;
378
379     HepVector a = mdcTrk->helix();
380     HepSymMatrix Ea = mdcTrk->err();
381     HepPoint3D point0(0.,0.,0.); // the initial point for MDC reconstruction
382     HepPoint3D IP(xorigin[0],xorigin[1],xorigin[2]);
383     VFHelix helixip(point0,a,Ea);
384     helixip.pivot(IP);
385     HepVector vecipa = helixip.a();
386     //helix P = (d0,phi0,kappa,z0,tan(lambda))
387     double Rvxy0=fabs(vecipa[0]); //the nearest distance to IP in xy plane
388     double Rvz0=vecipa[3]; //the nearest distance to IP in z direction
389     double Rvphi0=vecipa[1];
390     m_rvxy0=Rvxy0;
391     m_rvz0=Rvz0;
392     m_rvphi0=Rvphi0;
393
394     m_tuple1->write();
395     // if(fabs(z0) >= m_vz0cut) continue;
396     // if(fabs(Rxy) >= m_vr0cut) continue;
397
398     if(fabs(Rvz0) >= 10.0) continue;
399     if(fabs(Rvxy0) >= 1.0) continue;
400
401     iGood.push_back(i);
402     ncharge += mdcTrk->charge();
403 }
404 // Finish Good Charged Track Selection
405 //
406 int nGood = iGood.size();
407 log << MSG::DEBUG << "nGood, totcharge = " << nGood << " , " << ncharge << endl;
408 if((nGood != 2) || (ncharge!=0)){
409     return StatusCode::SUCCESS;
410 }
```

```
350 IVertexDbSvc* vtxsvc;
351 Gaudi::svcLocator()->service("VertexDbSvc", vtxsvc);
352 if(vtxsvc->isVertexValid()){
353     double* dbv = vtxsvc->PrimaryVertex();
354     double* vv = vtxsvc->SigmaPrimaryVertex();
355     // HepVector dbv = m_reader.PrimaryVertex(runNo);
356     // HepVector vv = m_reader.SigmaPrimaryVertex(runNo);
357     xorigin.setX(dbv[0]);
358     xorigin.setY(dbv[1]);
359     xorigin.setZ(dbv[2]);
360 }
```

- IP: 初始顶点, 从数据库读取。
- 径迹需要限制在一个小范围内 (排除宇宙线与束流本底)
- 电荷守恒以及本过程中带电径迹

数量要求

Return SUCCESS:排除这个事例,
中断这个事例的分析。

光子筛选



```
413 Vint iGam;
414 iGam.clear();
415 for(int i = evtRecEvent->totalCharged(); i < evtRecEvent->totalTracks(); i++) {
416     EvtRecTrackIterator itTrk=evtRecTrkCol->begin() + i;
417     if(!(*itTrk->isEmcShowerValid()) continue;
418     RecEmcShower *emcTrk = (*itTrk->emcShower();
419     Hep3Vector emcpos(emcTrk->x(), emcTrk->y(), emcTrk->z());
420     // find the nearest charged track
421     double dthe = 200.;
422     double dphi = 200.;
423     double dang = 200.;
424     for(int j = 0; j < evtRecEvent->totalCharged(); j++) {
425         EvtRecTrackIterator jtTrk = evtRecTrkCol->begin() + j;
426         if(!(*jtTrk->isExtTrackValid()) continue;
427         RecExtTrack *extTrk = (*jtTrk->extTrack();
428         if(extTrk->emcVolumeNumber() == -1) continue;
429         Hep3Vector extpos = extTrk->emcPosition();
430         // double ctht = extpos.cosTheta(emcpos);
431         double angd = extpos.angle(emcpos);
432         double thed = extpos.theta() - emcpos.theta();
433         double phid = extpos.deltaPhi(emcpos);
434         thed = fmod(thed+CLHEP::twopi+CLHEP::twopi+pi, CLHEP::twopi) - CLHEP::pi;
435         phid = fmod(phid+CLHEP::twopi+CLHEP::twopi+pi, CLHEP::twopi) - CLHEP::pi;
436         if(angd < dang){
437             dang = angd;
438             dthe = thed;
439             dphi = phid;
440         }
441     }
442     if(dang>=200) continue;
443     double eraw = emcTrk->energy();
444     dthe = dthe * 180 / (CLHEP::pi);
445     dphi = dphi * 180 / (CLHEP::pi);
446     dang = dang * 180 / (CLHEP::pi);
447     m_dthe = dthe;
448     m_dphi = dphi;
449     m_dang = dang;
450     m_eraw = eraw;
451     m_tuple2->write();
452     if(eraw < m_energyThreshold) continue;
453     if(fabs(dang) < m_gammaAngleCut) continue;
454 }
455 iGam.push_back(i);
456 }
457
458 //
459 // Finish Good Photon Selection
460 //
461 int nGam = iGam.size();
462
463 log << MSG::DEBUG << "num Good Photon " << nGam
464 if(nGam<2){
465     return StatusCode::SUCCESS;
466 }
467 Ncut2++;
```

```
611 Vp4 pGam;
612 pGam.clear();
613 for(int i = 0; i < nGam; i++) {
614     EvtRecTrackIterator itTrk = evtRecTrkCol->begin() + iGam[i];
615     RecEmcShower* emcTrk = (*itTrk->emcShower();
616     double eraw = emcTrk->energy();
617     double phi = emcTrk->phi();
618     double the = emcTrk->theta();
619     HepLorentzVector ptrk;
620     ptrk.setPx(eraw*sin(the)*cos(phi));
621     ptrk.setPy(eraw*sin(the)*sin(phi));
622     ptrk.setPz(eraw*cos(the));
623     ptrk.setE(eraw);
624
625 // ptrk = ptrk.boost(-0.011,0,0); // boost to cms
626
627 pGam.push_back(ptrk);
628 }
```

光子四动
量存储

对于i光子，循环所有带电径迹，找到i光子与所有带电径迹的最小夹角。一般设置最小夹角大于某个阈值，用于排除韧致辐射产生的shower。

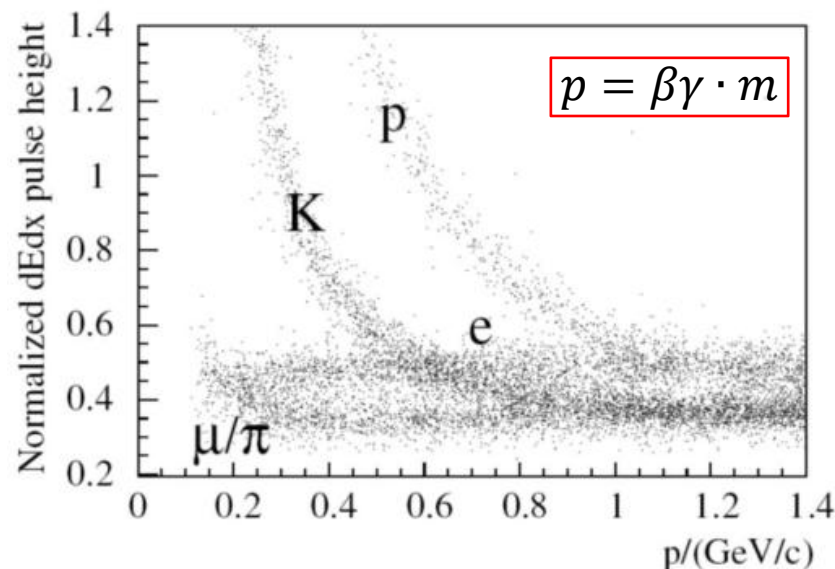
```
declareProperty("EnergyThreshold", m_energyThreshold=0.04);
68 declareProperty("GammaAngleCut", m_gammaAngleCut=20.0);
```

对于光子进行进一步筛选

对于光子的要求通常大于等于n。因为EMC很“脏乱”，会有各种shower，需要后续进一步筛选。在这里主要是由于我们筛选事例的末态至少会有两个光子。 27

```

472 //
473 // check dedx information
474 //
475 //
476
477 if(m_checkDedx == 1) {
478     for(int i = 0; i < nGood; i++) {
479         EvtRecTrackIterator itTrk = evtRecTrkCol->begin() + iGood[i];
480         if(!(*itTrk)->isMdcTrackValid()) continue;
481         if(!(*itTrk)->isMdcDedxValid()) continue;
482         RecMdcTrack* mdcTrk = (*itTrk)->mdcTrack();
483         RecMdcDedx* dedxTrk = (*itTrk)->mdcDedx();
484         m_ptrk = mdcTrk->p();
485
486         m_chie = dedxTrk->chiE();
487         m_chimu = dedxTrk->chiMu();
488         m_chipi = dedxTrk->chiPi();
489         m_chik = dedxTrk->chiK();
490         m_chip = dedxTrk->chiP();
491         m_ghit = dedxTrk->numGoodHits();
492         m_thit = dedxTrk->numTotalHits();
493         m_probPH = dedxTrk->probPH();
494         m_normPH = dedxTrk->normPH();
495         m_tuple7->write();
496     }
497 }
    
```



测量带电粒子于MDC中的电离能损来鉴别粒子：

$$-\left\langle \frac{dE}{dx} \right\rangle = K z^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{\max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right]$$

$$\chi = \frac{dE/dx(\text{测量}) - dE/dx(\text{期望})}{\sigma(dE/dx)}$$

低动量区： $-\frac{dE}{dx} \propto \frac{1}{\beta^2}$ ，由漂移室得到粒子动量，再根据dEdx信息可实现粒子鉴别。

检查TOF信息

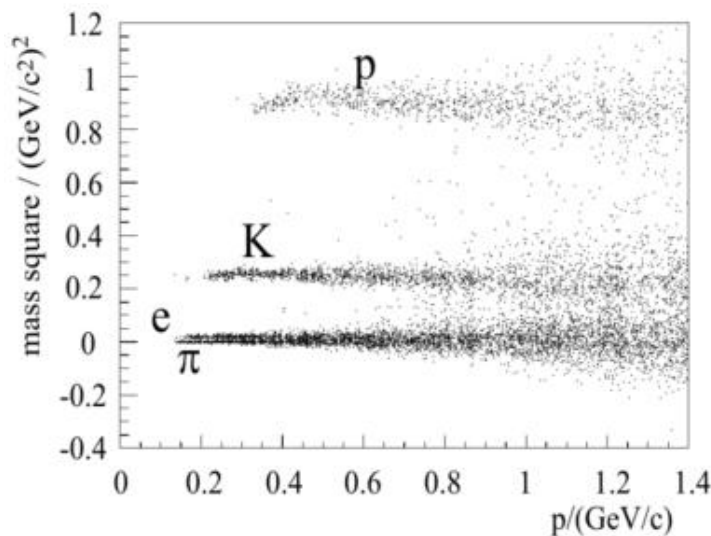


```
500 // check TOF information
501 //
502
503 if(m_checkTof == 1) {
504     for(int i = 0; i < nGood; i++) {
505         EvtRecTrackIterator itrk = evtRecTrkCol->begin() + iGood[i];
506         if(!(*itrk)->isMdcTrackValid()) continue;
507         if(!(*itrk)->isTofTrackValid()) continue;
508
509         RecMdcTrack * mdcTrk = (*itrk)->mdcTrack();
510         SmartRefVector<RecTofTrack> tofTrkCol = (*itrk)->tofTrack();
511
512         double ptrk = mdcTrk->p();
513         SmartRefVector<RecTofTrack>::iterator iter_tof = tofTrkCol.begin();
514         for(; iter_tof != tofTrkCol.end(); iter_tof++) {
515             TofHitStatus *status = new TofHitStatus;
516             status->setStatus((*iter_tof)->status());
517             if(!status->is_barrel()){ //endcap
518                 if( !(status->is_counter()) ) continue; // ?
519                 if( status->layer()!=0 ) continue; //layer1
520                 double path=(*iter_tof)->path(); // ?
521                 double tof = (*iter_tof)->tof();
522                 double ph = (*iter_tof)->ph();
523                 double rhit = (*iter_tof)->zrhithit();
524                 double qual = 0.0 + (*iter_tof)->quality();
525                 double cntr = 0.0 + (*iter_tof)->tofID();
526                 double texp[5];
527                 for(int j = 0; j < 5; j++) {
528                     double gb = ptrk/xmass[j];
529                     double beta = gb/sqrt(1+gb*gb);
530                     texp[j] = 10 * path /beta/velc;
531                 }
532                 m_cntr_etof = cntr;
533                 m_ptot_etof = ptrk;
534                 m_ph_etof = ph;
535                 m_rhit_etof = rhit;
536                 m_qual_etof = qual;
537                 m_te_etof = tof - texp[0];
538                 m_tmu_etof = tof - texp[1];
539                 m_tpi_etof = tof - texp[2];
540                 m_tk_etof = tof - texp[3];
541                 m_tp_etof = tof - texp[4];
542                 m_tuple8->write();
543             }
544             else { //barrel
545                 if( !(status->is_counter()) ) continue; // ?
546                 if(status->layer()==1) { //layer1
```

```
546         if(status->layer()==1) { //layer1
547         if(status->layer()==2) { //layer2
```

TOF的桶部分为两层

- 测量带电粒子从IP点至TOF的飞行时间。
$$t = L/c\beta$$
- 与dE/dx鉴别类似，动量与飞行时间实现



$$\chi = \frac{T(\text{测量}) - T(\text{期望})}{\sigma(T)}$$


```
ParticleID *pid = ParticleID::instance();
for(int i = 0; i < nGood; i++) {
    EvtRecTrackIterator itTrk = evtRecTrkCol->begin() + iGood[i];
    // if(pid) delete pid;
    pid->init();
    pid->setMethod(pid->methodProbability());
    pid->setMethod(pid->methodLikelihood()); //for Likelihood Method

    pid->setChiMinCut(4);
    pid->setRecTrack(*itTrk);
    pid->usePidSys(pid->useDedx() | pid->useTof1() | pid->useTof2() | pid->useTofE()); // for
    sion < 7.0.0
    pid->usePidSys(pid->useDedx() | pid->useTofCorr()); // use PID sub-system
    pid->identify(pid->onlyPionKaonProton()); // seperater Pion/Kaon/Proton
    // pid->identify(pid->onlyPion());
    // pid->identify(pid->onlyKaon());
    pid->calculate();
    if(!(pid->IsPidInfoValid())) continue;
    RecMdcTrack* mdcTrk = (*itTrk)->mdcTrack();
    m_ptrk_pid = mdcTrk->p();
    m_cost_pid = cos(mdcTrk->theta());
    m_dedx_pid = pid->chiDedx(2);
    m_tof1_pid = pid->chiTof1(2);
    m_tof2_pid = pid->chiTof2(2);
    m_prob_pid = pid->probPion();
    m_tuple11->write();

    if((pid->probPion() < pid->probProton()) || (pid->probPion() < pid->probKaon())) continue;
```

表明调用的粒子鉴别方法

dEdx与TOF联合鉴别同时设定备选的鉴别粒子种类

```
if(npip*npim != 1) return SUCCESS;
Ncut3++;
```

由鉴定后的结果进行进一步筛选

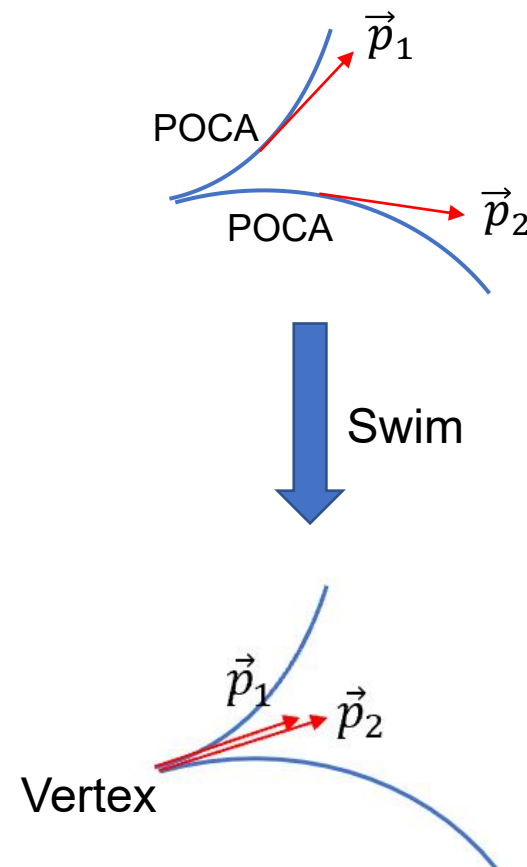
```
743 RecMdcKalTrack *pipTrk = (*(evtRecTrkCol->begin()+ipip[0]))->mdcKalTrack();
744 RecMdcKalTrack *pimTrk = (*(evtRecTrkCol->begin()+ipim[0]))->mdcKalTrack();
745
746 WTrackParameter wvpipTrk, wvpimTrk;
747 wvpipTrk = WTrackParameter(mpi, pipTrk->getZHelix(), pipTrk->getZError());
748 wvpimTrk = WTrackParameter(mpi, pimTrk->getZHelix(), pimTrk->getZError());
...
```

PID之后，已确定粒子类型，需赋予径迹质量。RecMdcTrack升级为RecMdcKalTrack。可拿到**径迹参数**，用于顶点拟合。

```
756 //  
757 // Test vertex fit  
758 //  
759  
760 HepPoint3D vx(0., 0., 0.);  
761 HepSymMatrix Evx(3, 0);  
762 double bx = 1E+6;  
763 double by = 1E+6;  
764 double bz = 1E+6;  
765 Evx[0][0] = bx*bx;  
766 Evx[1][1] = by*by;  
767 Evx[2][2] = bz*bz;  
768  
769 VertexParameter vxpar;  
770 vxpar.setVx(vx);  
771 vxpar.setEvx(Evx);  
772  
773 VertexFit* vtxfit = VertexFit::instance();  
774 vtxfit->init();  
775 vtxfit->AddTrack(0, wvpipTrk);  
776 vtxfit->AddTrack(1, wvpimTrk);  
777 vtxfit->AddVertex(0, vxpar, 0, 1);  
778 if(!vtxfit->Fit(0)) return SUCCESS;  
779 vtxfit->Swim(0);  
780  
781 WTrackParameter wpip = vtxfit->wtrk(0);  
782 WTrackParameter wpim = vtxfit->wtrk(1);
```

初始设置一个误差无穷大的顶点

- 末态径迹需要来自于同一个顶点
- 多数粒子寿命很短，衰变顶点与产生顶点“重合”。



```

790 if(m_test4C==1) {
791   HepLorentzVector ecms(0.034,0,0,3.097);
792   double chisq = 9999.;
793   int ig1 = -1;
794   int ig2 = -1;
795   for(int i = 0; i < nGam-1; i++) {
796     RecEmcShower *g1Trk = (*(evtRecTrkCol->begin()+iGam[i]))->emcShower();
797     for(int j = i+1; j < nGam; j++) {
798       RecEmcShower *g2Trk = (*(evtRecTrkCol->begin()+iGam[j]))->emcShower();
799       kmfit->init();
800       kmfit->AddTrack(0, wpip);
801       kmfit->AddTrack(1, wpim);
802       kmfit->AddTrack(2, 0.0, g1Trk);
803       kmfit->AddTrack(3, 0.0, g2Trk);
804       kmfit->AddFourMomentum(0, ecms);
805       bool oksq = kmfit->Fit();
806       if(oksq) {
807         double chi2 = kmfit->chisq();
808         if(chi2 < chisq) {
809           chisq = chi2;
810           ig1 = iGam[i];
811           ig2 = iGam[j];
812         }
813       }
814     }
815   }
816   if(chisq < 200) {
817     RecEmcShower *g1Trk = (*(evtRecTrkCol->begin()+ig1))->emcShower();
818     RecEmcShower *g2Trk = (*(evtRecTrkCol->begin()+ig2))->emcShower();
819     kmfit->init();
820     kmfit->AddTrack(0, wpip);
821     kmfit->AddTrack(1, wpim);
822     kmfit->AddTrack(2, 0.0, g1Trk);
823     kmfit->AddTrack(3, 0.0, g2Trk);
824     kmfit->AddFourMomentum(0, ecms);
825     bool oksq = kmfit->Fit();
826     if(oksq) {
827       HepLorentzVector ppi0 = kmfit->pfit(2) + kmfit->pfit(3);
828       m_mpi0 = ppi0.m();
829       m_ch1 = kmfit->chisq();
830       m_tuple4->write();
831       Ncut4++;
832     }
833   }
834 }
835 }

```

4C

```

842 if(m_test5C==1) {
843   // double ecms = 3.097;
844   HepLorentzVector ecms(0.034,0,0,3.097);
845   double chisq = 9999.;
846   int ig1 = -1;
847   int ig2 = -1;
848   for(int i = 0; i < nGam-1; i++) {
849     RecEmcShower *g1Trk = (*(evtRecTrkCol->begin()+iGam[i]))->emcShower();
850     for(int j = i+1; j < nGam; j++) {
851       RecEmcShower *g2Trk = (*(evtRecTrkCol->begin()+iGam[j]))->emcShower();
852       kmfit->init();
853       kmfit->AddTrack(0, wpip);
854       kmfit->AddTrack(1, wpim);
855       kmfit->AddTrack(2, 0.0, g1Trk);
856       kmfit->AddTrack(3, 0.0, g2Trk);
857       kmfit->AddResonance(0, 0.135, 2, 3);
858       kmfit->AddFourMomentum(1, ecms);
859       if(!kmfit->Fit(0)) continue;
860       if(!kmfit->Fit(1)) continue; //if(!kmfit->Fit()) continue;
861       bool oksq = kmfit->Fit();
862       if(oksq) {
863         double chi2 = kmfit->chisq();
864         if(chi2 < chisq) {
865           chisq = chi2;
866           ig1 = iGam[i];
867           ig2 = iGam[j];
868         }
869       }
870     }
871   }

```

5C

通过置信度检验判断拟合情况, χ^2 越小, 拟合越好。



- 于initialize()中定义tree与tuple
- 变量赋值后，需要通过write()将其写入.root中
- Write()后，tuple中的m_变量失效

```
149 NTuplePtr nt5(ntupleSvc(), "FILE1/fit5c");
150 if ( nt5 ) m_tuple5 = nt5;
151 else {
152     m_tuple5 = ntupleSvc()->book ("FILE1/fit5c", CLID_ColumnwiseTuple, "ks N-Tuple example");
153     if ( m_tuple5 ) {
154         status = m_tuple5->addItem ("chi2", m_chi2);
155         status = m_tuple5->addItem ("mrh0", m_mrh0);
156         status = m_tuple5->addItem ("mrhp", m_mrhp);
157         status = m_tuple5->addItem ("mrhm", m_mrhm);
158     }
159     else {
160         log << MSG::ERROR << "      Cannot book N-tuple:" << long(m_tuple5) << endmsg;
161         return StatusCode::FAILURE;
162     }
163 }
```

变量的类型于.h中声明

```
894 m_chi2 = kmfit->chisq();
895 m_mrh0 = prho0.m();
896 m_mrhp = prhop.m();
897 m_mrhm = prhom.m();
898 double eg1 = (kmfit->pfit(2)).e();
899 double eg2 = (kmfit->pfit(3)).e();
900 double fcos = abs(eg1-eg2)/ppi0.rho();
901 m_tuple5->write();
---
```

finalize()



- dst中的所有事例遍历完后，执行 finalize()
- 用于展示cut flow
- 也可作为分析程序运行成功的一个标志

```

923 StatusCode Rhopi::finalize() {
924     cout<<"total number: " << Ncut0<<endl;
925     cout<<"nGood==2, nCharge==0: " << Ncut1<<endl;
926     cout<<"nGam>=2: " << Ncut2<<endl;
927     cout<<"Pass Pid: " << Ncut3<<endl;
928     cout<<"Pass 4C: " << Ncut4<<endl;
929     cout<<"Pass 5C: " << Ncut5<<endl;
930     cout<<"J/psi->rho0 pi0: " << Ncut6<<endl;
931     MsgStream log(msgSvc(), name());
932     log << MSG::INFO << "in finalize()" << endmsg;
933     return StatusCode::SUCCESS;
934 }

```

```

407 log << MSG::DEBUG << "ngood, totcharge = " << nGood << " , " << nCharge << endreq;
408 if((nGood != 2) || (nCharge!=0)){
409     return StatusCode::SUCCESS;
410 }
411 Ncut1++;

```

```

463 log << MSG::DEBUG << "num Good Photon " << nGam << " , " << evtRecEvent->totalNeutral() << endreq;
464 if(nGam<2){
465     return StatusCode::SUCCESS;
466 }
467 Ncut2++;

```

```

718 int npip = ipip.size();
719 int npim = ipim.size();
720 if(npip*npim != 1) return SUCCESS;
721
722 Ncut3++;

```

Selection Criteria	Absolute Efficiency (%)	Relative Efficiency (%)
Good Charged Tracks and PID	66.74	—
Vertex Fit	65.22	97.73
Good Shower Selection	57.68	88.45
Kinematic Fit	48.39	83.89
$\chi^2_{sec} < 15$ and $L/\sigma_L > 2.0$	42.88	88.61
$\chi^2_{corr} < 70$	35.17	82.01
$ M_{\bar{p}\pi^+} - M_{\Lambda} < 8 MeV/c^2$	33.19	94.37
$M_{\bar{p}\pi^+}^{recoil} < 1.15 GeV/c^2$	32.92	99.19
π^0 Asymmetry	28.33	86.05
$\theta_{\Lambda, Shower} > 10^\circ$	24.13	85.19
$0.88 < M_n^{recoil} < 0.98 (GeV/c^2)$	21.09	87.39

一般跑分析作业运行成功时的标志：

```
ApplicationMgr      INFO Application Manager Stopped successfully
Total number:      20000
Nmc:                20000
nGood>=2:          19599
npbar>=1 && npip>=1: 14303
nGam>=1:           13136
Vertex sel(ST):     12669
Npick(ST):          12669
nGood>=6:           1302
np>=1 && npim>=1:    1195
Vertex sel(DT):     1134
nep>=1 && nem>=1:    374
ApplicationMgr      INFO Application Manager Finalized successfully
ApplicationMgr      INFO Application Manager Terminated successfully
```


- BESIII 探测器
 - 各个子探测器(MDC, TOF, EMC, MUC)的主要功能
- 分析算法的主要框架：
 - initialize()
 - execute() —— event by event
 - Finalize()
- execute() 主要包含：
 - 带电径迹与shower读取
 - 粒子鉴别(PID)
 - 顶点拟合
 - 运动学拟合

谢谢！