MCFast

Fast Detector Simulation

from F.N.A.L.

Lynn Garren
Dec. 6, 1995

MCFast Simulation Package

Mcfast is a fast tracking simulation developed by the simulation group in the Fermilab Computing Division

Authors:

P. Avery, UFlorida (at FNAL in '94)
A. Boehnlein
L. Garren
P. Lebrun
P. McBride
M. Procario, CMU (FNAL '95)
K. Sterner (FNAL/UPenn)
T. Wenaus, LLNL (FNAL summer '95)
J. Yarba

Information available: http://fnpspa.fnal.gov/simulation.html

Future Experiments–– beyond 2000

A second generation B experiment must go beyond the measurement of CP asymmetry with $B^0_d \to \psi K^0_s$ and do a complete program of B physics:

- $B^0_d \to \pi^+ \pi^-$
- $B \to D^0 K$
- $B \to D^+ D^-$
- B_s and B baryon modes
- B_s Mixing
- B rare decays

The Tevatron will produce a large numbers of B hadrons:

$10^{10} - 10^{11}$ per year
expected with the Main Injector.

But there are large backgrounds to consider:

$$\sigma_{bb}/\sigma_{inel} \sim 10^{-3}$$

and many interesting final states have

branching ratios $\sim 10^{-5}$

B production at the Tevatron

Cross section for B production at the Tevatron (1.8 TeV) ~ 50 μb

Number of B^0_d produced @ 10^{32} in a Snowmass year (10^7 s) 3.75×10^{10}

Channels for observing CP asymmetry:

Branching Ratio $B^0_d \to \psi K^0_s$ 5.5×10^{-4}

$\to \mu^+\mu^-\pi^+\pi^-$ 2.2×10^{-5}

$B^0_d \to \psi K^0_s \to \mu^+\mu^-\pi^+\pi^-$ produced per Snowmass year 8.2×10^{5}

Branching Ratio $B^0_d \to \pi^+\pi^-$ ~ 10^{-5}

$B^0_d \to \pi^+\pi^-$ produced per Snowmass year ~ 4×10^{5}

Understanding the efficiency and backgrounds for channels such as $B^0_d \rightarrow \pi^+ \pi^-$ is important design issue for a dedicated B detector.

Which detector geometry is best?

We need a systematic way to compare the physics reach of various detector options:

Central (Solenoidal) Detectors ($|\eta| < 2.0$)
-- CDF or D0

Forward Collider detectors ($1.5 < \eta < 4.5$)
(with a forward Dipole) --LHB

Central dipole -- BCD

Combination

Stategy for Studies

Design a tool to make a fair comparison of different detector geometries.

-- should work at several energies with several event generators

-- simple but realistic representation of detector resolutions and multiple scattering

-- accurate model of B decays (and charm)

-- model interaction region and multiple interactions per crossing

Start by looking at signal/background in various "ideal" detector geometries and then study trigger and tagging efficiencies.

Requirements for B Physics simulation:

Flexible –– goal is to study various detector options

Fast –– need to simulate ~1M bb events to understand background rates in some channels

Need a tool to enable the study of:

- Acceptance
- Background rejection
- Momentum resolution
- Mass resolution
- Secondary vertex resolution
- Tagging
- Trigger

What about existing software?

GEANT: too slow

TRKERR(ASLUND) not designed for hadron colliders; besides it considers only solenoids

What is MCFast

evgen

- QQ

StdHep

MCFio

MCFast

- geometry
- vertexing
- tracking

an example

MCFAST is a fast tracking simulation which accurately reproduces momentum, mass, and vertex resolutions.

Original motivation to make fair comparisons between B experiments at the Tevatron.

Work begun 3 years ago.
Needed for SSC work....

Simulates both detector and analysis.

Background studies require many events.

I. A flexible but limited geometry read from ASCII files.

- Handles both central and forward geometries.
- useful for collider or fixed target experiments
- No coding of detector response required of the user.

II. Interfaced to a variety of event generators.

- Uses StdHep routines.
- Currently can handle Pythia, ISAJET and Herwig.
- Can use CLEO decay package to decay b and c hadrons from the above packages.

III. Accurate tracking simulations by finding correct covariance matrix.

- Includes effects of multiple scattering and detector efficiencies.

- Approach is similar to TRKERR and other available programs but more general.

    + Handles forward geometry as well as central.
    + Handles tracks from secondary decays.
    + Primary vertex can move event by event.
    + Multiple interations per crossing allowed.
    + Handles decays in flight, gamma conversions.

IV. Vertex fitting package

- Fits either primary or secondary vertices.
- Results of fits can be used a particles in subsequent fits.

V. 3–D graphics for detector design.

- Interactive 3–D graphics are available for studying detector as it is designed.

VI. Does generate smeared hits for occupancy studies.

VII. Simple algorithms for other detectors

- Muon identification
- Calorimetry
- Particle ID (primitive)

Fast enough to run through large background samples.

- 0.2 to 1.0 second per event depending on physics process simulated (using a high speed computer like a 150 MHz R4400)

Simulates analysis.

MCFAST is well optimized to do studies involving secondary vertices and mass combinations, which will dominate the study of B's at hadron coliders.

MCFAST Group

Paul Avery U. of Florida	– godfather: started work during sabbatical no longer active
Patty McBride FNAL	– pulling it all together
Kevin Sterner FNAL/U of Penn	– forward tracking
Julia Yarba FNAL	– geometry specification; calorimetry
Mike Procario FNAL/CMU	– validation, forward tracking, vertexing
Amber Boehnlein FNAL	– trigger, vertexing, graphics
Paul Lebrun FNAL	– I/O package
Lynn Garren	– interfacing the event generators

Approximately 4 FTE's working now, up from 2.5 last year.

evgen

command line interface

Isajet

Pythia/Jetset

Herwig

QQ

- b & c decays
- used by CDF
- user–controlled decay definitions
- user decay table

```
;
QQBAR    *UD*    2
QQBAR    *UC*    4
QQBAR    *UB*    5
QQBAR    *DU*    7
;
PARTICLE  B-     33   -1  5.280000 -1.0  0.0   0.000493
PARTICLE  B+     34   -1  5.280000  1.0  0.0   0.000493
PARTICLE  B0B     35   -1  5.280000  0.0  0.0   0.000486
PARTICLE  B0     36   -1  5.280000  0.0  0.0   0.000486
PARTICLE  ETAC    54   -1  2.979600  0.0  0.0   0.000000 \
                              0.010  2.960  3.000
;
;Pseudo 2-BODY charm states used in Inclusive B decays
;
PARTICLE CU2  291 -1 3.000  0.0 0.0 0.0  0.500  2.300  3.500
PARTICLE CU2B 292 -1 3.000  0.0 0.0 0.0  0.500  2.300  3.500
PARTICLE CD2  293 -1 3.000  1.0 0.0 0.0  0.500  2.300  3.500
PARTICLE CD2B 294 -1 3.000 -1.0 0.0 0.0  0.500  2.300  3.500
;
;Define some temporary particles as hidden so that they will
;never appear in the particle history. Only the daughters of
;the hidden particle will appear.
;HIDE CU2
;HIDE CU2B
;
;     PDG numbering scheme
;
PDG      B-        -521
PDG      B+         521
PDG      B0B        -511
PDG      B0         511
;
PARITY  RHO-   -1
;
; decay section goes here.......
;
USERFILE
```

```
DECAY  B-
;                                b --> c  (l nu)
;                                Sum = 24.5%
;                                   D X l   nu  10.4% * 2
;                                   D X tau nu   2.7%
;                                   LC X l  nu   0.5% * 2
;
;                          D* angular dist = 1 + 1.25*COS**2
HELICITY 1.0  0  0  1
CHANNEL 52 0.0115 NUEB  E-   D*0
HELICITY 1.0  0  0 -1
CHANNEL 52 0.0115 NUEB  E-   D*0
HELICITY 1.0  0  0  0
CHANNEL 53 0.0260 NUEB  E-   D*0
CHANNEL 51 0.0180 NUEB  E-   D0
CHANNEL 51 0.0010 NUEB  E-   3P00
CHANNEL 51 0.0010 NUEB  E-   3P10
CHANNEL 51 0.0110 NUEB  E-   1P10
CHANNEL 51 0.0110 NUEB  E-   3P20
CHANNEL  1 0.0130 NUEB  E-   CUX2
;
CHANNEL  0 0.0100 DS-   D+  PI-
CHANNEL  0 0.0050 DS-   D0  PI0
CHANNEL  0 0.0100 DS*-  D+  PI-
CHANNEL  0 0.0050 DS*-  D0  PI0
;
CHANNEL  0 0.0055 DS-   D+  PI-  PI0
CHANNEL  0 0.0055 DS-   D0  PI-  PI+
CHANNEL  0 0.0014 DS-   D0  PI0  PI0
CHANNEL  0 0.0055 DS*-  D+  PI-  PI0
CHANNEL  0 0.0055 DS*-  D0  PI-  PI+
CHANNEL  0 0.0014 DS*-  D0  PI0  PI0
;                                B --> Charm baryons + (d u=)
;                                B --> Charm baryons + (s u=)
;                                Sum = 6.0%
CHANNEL  9 0.0550 LC-PB *DU*
CHANNEL  9 0.0050 LC-PB *SU*
ENDDECAY
```

```
DECAY  B+
;
CHANNEL  0 0.0094 DS+   D0B
HELICITY   1.0    0    0
CHANNEL  0 0.0090 DS+   D*B
HELICITY   1.0    0    0
CHANNEL  0 0.0094 DS*+  D0B
HELICITY   0.333   1   -1
HELICITY   0.334   0    0
HELICITY   0.333  -1    1
CHANNEL  0 0.0174 DS*+  D*B
;
CHANNEL  0 0.0055 DS+   D-    PI+  PI0
CHANNEL  0 0.0055 DS+   D0B  PI+  PI-
CHANNEL  0 0.0014 DS+   D0B  PI0  PI0
;
CHANNEL  0 0.0002 ETAC  K+
HELICITY   1.0    0    0
CHANNEL  0 0.0005 ETAC  K*+
CHANNEL  0 0.0001 ETAC  K0    PI+
CHANNEL  0 0.0001 ETAC  K+    PI0
CHANNEL  0 0.0001 ETAC  K+    PI-   PI+
CHANNEL  0 0.0001 ETAC  K0    PI0   PI+
CHANNEL  0 0.0001 ETAC  K+    PI0   PI0
;
CHANNEL  0 0.0020 D-   RHO+  PI+
CHANNEL  0 0.0020 D-   PI0   PI+   PI+
;
CHANNEL  0 0.0005 D*B  RHO+  PI0
CHANNEL  0 0.0005 D*B  RHO+  PI0   PI0
;
;                          b --> c  (d u=)
;                          Inclusive unknowns
;                                   43.73%
CHANNEL  9 0.0600 *UD*  *UC*
CHANNEL  8 0.3300 *UD*  *UC*
CHANNEL  0 0.0473 *UD*  *UC*
ENDDECAY
```

Sample user decay table

```
; turn off pi0 decay
;
DECAY  PI0
;
ENDDECAY
;
; exclusive B– decay
;
DECAY  B–
CHANNEL  0  1.0  D*0  PI–
ENDDECAY
```

StdHep

Motivated by introduction of HEPEVT common block and demise of pytios

Implementation of standard particle numbering scheme not consistent, or lacking entirely

StdHep provides a consistent interpretation of the standard numbering scheme for Pythia/Jetset, Isajet, Herwig, and QQ.

Utilitities to get information from HEPEVT

I/O utilities

- zebra
- MCFio

MCFast expects generator level information in standard HEP event format.

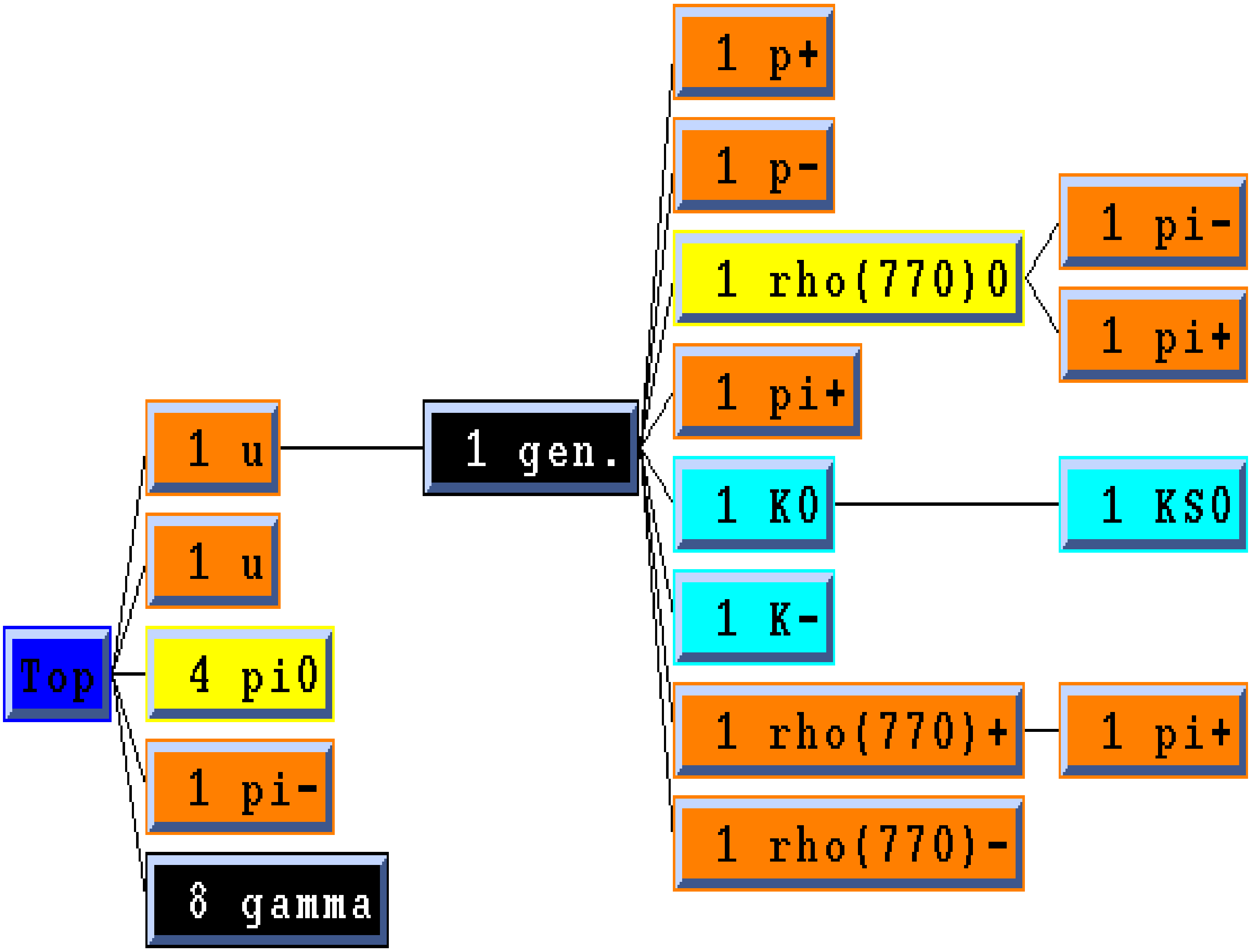
Gauge and Higgs Bosons						
Particle	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
g	9	21	21	-13	21 (9)	21 (9)
γ	10	22	22	1	22	22
Z^0	90	23	23	2	23	23
W^+	80	24	24	3,4	24	24
H_1^0	81 (H^0)	25 (H^0)	25		25	25
η^0_{tech}		38				27
LQ^0		39				28
R^0		40				29
Z_2^0		32 (Z'^0)	32 (Z'^0)			30
W_2^+		34 (W'^+)				31
H_2^0	82 $(H_1^0)^{NOTE}$	35 (H'^0)			35 (H_2^0)	32
H_2^+	86 (H^+)	37 (H^+)			37 (H^+)	33
H_2^{++}	88 (H_1^{++})					34
Z_3^0		33 (Z''^0)				35
H_3^0	83 $(H_2^0)^{NOTE}$	36 (A^0)			36 (H_3^0)	36
H_3^+	87 (H_2^+)					37
H_3^{++}	89 (H_2^{++})					38
H_4^0	84 $(H_3^0)^{NOTE}$					39
H_5^0	85 (H_4^0)					40

NOTE: In Isajet 7.06, H_1^0, H_2^0, and H_3^0 are removed.

Charmed Mesons						
Meson	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
ψ^0_{diffr}		440				440
$\eta_c(1S)$	440	441	441	54	441	441
$\chi_{c0}(1P)$		10441	10441 (χ)	110	10441	10441
$\eta_c(2S)$						20441
J/ψ	441	443	443	94	443	443
$h_c(1P)$		10443		215		10443
$\chi_{c1}(1P)$		20443	10441 (χ)	111	10443	20443
$\psi(2S)$		30443 (ψ')	20443 (ψ')	219	20443	30443
$\psi(3770)$			30443 (ψ'')	211	30443	40443
$\psi(4040)$				212	40443	50443
$\psi(4160)$				213	50443	60443
$\psi(4415)$				214	60443	70443
$\chi_{c2}(1P)$		445	10441 (χ)	112	445	445

Charmed Baryons							
Baryon	quark content	Isajet 7.13	Jetset 7.4	Herwig 5.8	QQ 9.1	1994 PDG	StdHep 3.01
Λ_c^+	duc	2140	4122	4122	123,124	4122	4122
Σ_c^{++}	uuc	1140	4222	4222	141,142	4222	4222
Σ_c^+	duc	1240	4212	4212	131,132	4212	4212
Σ_c^0	ddc	2240	4112	4112	147,148	4112	4112
Σ_c^{*++}	uuc	1141	4224	4224	173,174		4224
Σ_c^{*+}	duc	1241	4214	4214	163,164		4214
Σ_c^{*0}	ddc	2241	4114	4114	179,180		4114
Ξ_c^+	usc	3140	4232	4232	125,126	4322	4232
Ξ_c^0	dsc	3240	4132	4132	127,128	4312	4132
$\Xi_c^{\prime +}$	usc	1340	4322	4322 (SS+)	133,134		4322
$\Xi_c^{\prime 0}$	dsc	2340	4312	4312 (SS0)	135,136		4312
Ξ_c^{*+}	usc	1341	4324	4324 (SS*+)	165,166		4324
Ξ_c^{*0}	dsc	2341	4314	4314 (SS*0)	167,168		4314
Ω_c^0	ssc	3340	4332	4332	153,154	4332	4332
Ω_c^{*0}	ssc	3341	4334	4334 (T^{*0})	185,186		4334
Ξ_{cc}^+	dcc	2440	4412		157,158		4412
Ξ_{cc}^{++}	ucc	1440	4422		155,156		4422
Ξ_{cc}^{*+}	dcc	2441	4414		189,190		4414
Ξ_{cc}^{*++}	ucc	1441	4424		187,188		4424
Ω_{cc}^+	scc	3440	4432		159,160		4432
Ω_{cc}^{*+}	scc	3441	4434		191,192		4434
Ω_{ccc}^{*++}	ccc	4441	4444		199,200		4444

```
C
      integer nmxhep
      parameter (nmxhep=4000)
      common/HEPEVT/nevhep,nhep,isthep(nmxhep),
     & idhep(nmxhep),jmohep(2,nmxhep),jdahep(2,nmxhep),
     & phep(5,nmxhep),vhep(4,nmxhep)
      integer nevhep,nhep,isthep,idhep,jmohep,jdahep
      double precision phep,vhep
      save /HEPEVT/
C...      nevhep       – event number
C...      nhep         – number of entries in this event
C...      isthep(..)   – status code
C...      idhep(..)    – particle ID, P.D.G. standard
C...      jmohep(1,..) – position of mother particle in list
C...      jmohep(2,..) – position of second mother particle
C...      jdahep(1,..) – position of first daughter in list
C...      jdahep(2,..) – position of last daughter in list
C...      phep(1,..)   – x momentum in GeV/c
C...      phep(2,..)   – y momentum in GeV/c
C...      phep(3,..)   – z momentum in GeV/c
C...      phep(4,..)   – energy in GeV
C...      phep(5,..)   – mass in GeV/c**2
C...      vhep(1,..)   – x vertex position in mm
C...      vhep(2,..)   – y vertex position in mm
C...      vhep(3,..)   – z vertex position in mm
C...      vhep(4,..)   – production time in mm/c
C
C      common/hepev2/nmulti,jmulti(nmxhep)
      integer nmulti,jmulti
      save /hepev2/
C...      nmulti       – number of interactions in the list
C...      jmulti(..)   – multiple interaction number
C
```



Event tree structure => C++

StdHep Translation Routines

hep2geant
Put undecayed HEPEVT particles into the Geant KINE stack. Particles unknown to GEANT are entered as geantino's.

hwghep(mconv)
Translate to or from Herwig event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the Herwig common blocks

isahep(mconv)
Translate to or from Isajet event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the Isajet common blocks

lunhep(mconv)
Translate to or from Jetset event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the Jetset common blocks

qqhep(mconv)
Translate to or from QQ event format.

mconv	=	1	fill the HEPEVT common block
	=	2	fill the QQ common blocks
	=	3	fill the QQ common blocks for decay of "stable" particles

StdHep Translation Routines

gtran(id,mconv)

Translate to or from Geant particle id numbering.

gtran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from Geant to StdHep numbering
	=	2	convert from StdHep to Geant numbering

istran(id,mconv)

Translate to or from Isajet particle id numbering.

istran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from Isajet to StdHep numbering
	=	2	convert from StdHep to Isajet numbering

lutran(id,mconv)

Translate to or from Jetset particle id numbering.

lutran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from Jetset to StdHep numbering
	=	2	convert from StdHep to Jetset numbering

StdHep Translation Routines

hwtran(id,mconv)

Translate to or from Herwig particle id numbering.

hwtran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from Herwig to StdHep numbering
	=	2	convert from StdHep to Herwig numbering

qqtran(id,mconv)

Translate to or from QQ particle id numbering.

qqtran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from QQ to StdHep numbering
	=	2	convert from StdHep to QQ numbering

pdgtran(id,mconv)

Translate to or from PDG particle id numbering.

pdgtran	=		requested particle ID
id	=		particle ID to translate
mconv	=	1	convert from PDG to StdHep numbering
	=	2	convert from StdHep to PDG numbering

StdHep Translation Routines

pdgrdtb
Read particle mass and width info from the PDG table.
Fill the PDGTBL and STDTBL common blocks.

stdflhwxsec
Fill common block STDCM1 from Herwig.
STDCM1 contains center−of−mass energy, cross−section, random number seeds, number of events to be generated, number of events generated, and number of events written.

stdflisxsec
Fill common block STDCM1 from Isajet.

stdflpyxsec(n1)
Fill common block STDCM1 from Pythia.

n1 = number of events to be generated

hptrlsth Print full translation list for Herwig.
hptrlsti Print full translation list for Isajet.
hptrlstj Print full translation list for Jetset.
hptrlstq Print full translation list for QQ.

Basic StdHep Utility Routines

heplst(mlst)
List an event from the HEPEVT common block.

mlst = 1 print information from HEPEVT, excluding the vertex
= 2 print all information from HEPEVT, including the vertex

hepnam(id,chau)
Return the particle name as a character string

chau = 20 character particle name
id = particle ID

hepcmp(id)
Compress the StdHep particle ID for use in mass and decay arrays of size 1600. Also check to be sure the given ID is defined.

hepcmp = compressed particle ID
id = StdHep particle ID

hepchg(id)
Return an integer charge which is 3 times the charge of the particle.

hepchg = 3 times the charge of the particle
id = StdHep particle ID

stdchg(id)
Return the charge of particle id as a real number.

stdchg = the charge of the particle
id = StdHep particle ID

Basic StdHep Utility Routines

stdexpname(id,lsz)

Return an expanded particle name for particle id.
Fill and pad the variable with empty space if necessary.

stdexpname	=	expanded particle name
id	=	StdHep particle ID
lsz	=	length of the character variable

stdlst

List the particles known to StdHep.

StdHep Routines for QQ Decay

stddecayqq
Search the HEPEVT list for particles which QQ can decay. One by one, decay these particles and add the results to the HEPEVT common block.

stdqqdcy(it)
Decay particle it via QQ and add the results to the HEPEVT common block.

it = the index for this particle within HEPEVT

stdzeroqq
Zero elements of QQ common blocks before performing the QQ decays. This routine is called by stddecayqq.

stdqqadd(ihep)
Add tracks generated by QQ to the HEPEVT common block.

ihep = the first QQ particle corresponds to HEPEVT particle ihep

stdqquset
Call QQ initialization routines. This routine must be called by the user at initialization time.

stdsort
Sort the HEPEVT common block by daughters.
This routine is called by qqhep.

StdHep Utility Routines

stdquarks(ip,kq1,kq2,kq3,kqx,kqj)
Get the constitutent quarks of particle ip.
All numbers are positive.

ip	=	index of particle in HEPEVT
kq1	=	quark L of IJKLM
kq2	=	quark K of IJKLM
kq3	=	quark J of IJKLM
kqx	=	I of IJKLM
kqj	=	2*Jspin + 1

stdcquarks(ip,kq1,kq2,kq3,kqx,kqj)
Get the signed constitutent quarks of particle ip.

ip	=	index of particle in HEPEVT
kq1	=	quark L of IJKLM
kq2	=	quark K of IJKLM
kq3	=	quark J of IJKLM
kqx	=	I of IJKLM
kqj	=	2*Jspin + 1

stdspin(ip,xjsp)
Return the J spin of particle ip as a real number.

ip	=	index of particle in HEPEVT
xjsp	=	J spin

StdHep Utility Routines

stdquarklst(iq,nqrk,lsize,lqrk)
Search the HEPEVT common block for all mesons and baryons with quark content iq. Fill array lqrk(lsize).

iq = quark identification
nqrk = number of particles found
lsize = length of the array
lqrk = array of particle indices in HEPEVT

stdparentlst(ip,nprnt,lsize,lprnt)
Search the HEPEVT common block for all ancestors of particle ip. Fill lprnt(lsize).

ip = index of particle in HEPEVT
nprnt = number of particles found
lsize = length of the array
lprnt = array of particle indices in HEPEVT

stdparent(ip,ipc,lyn)
Check to see if particle ipc is a parent of particle ip.

ip = index of particle in HEPEVT
ipc = index of prospective parent in HEPEVT
lyn = true or false

stdquarkcnt(ip,iq,iret,ipar)
Look to see if particle it or any of its parents contain quark iq.

ip = index of particle in HEPEVT
iq = quark identification
iret = 0 no match found
= 1 this particle contains quark iq
= 2 parent ipar contains quark iq
ipar = 0 unless iret is 2

StdHep Utility Routines

stddautrlst(ip,ndau,lsize,ldtr)

Search the HEPEVT common block for the immediate daughters of particle ip. Fill ldtr(lsize).

ip = index of particle in HEPEVT
ndau = number of particles found
lsize = length of the array
ldtr = array of particle indices in HEPEVT

stddscndlst(ip,ndsc,lsize,ldsc)

Search the HEPEVT common block for all the descendants of particle ip. Fill ldsc(lsize).

ip = index of particle in HEPEVT
ndsc = number of particles found
lsize = length of the array
ldsc = array of particle indices in HEPEVT

stdstdsclst(ip,ndsc,lsize,ldsc)

Search the HEPEVT common block for all the stable descendants of particle ip. Fill ldsc(lsize).

ip = index of particle in HEPEVT
ndsc = number of particles found
lsize = length of the array
ldsc = array of particle indices in HEPEVT

StdHep Utility Routines

stdchgdsclst(ip,ndsc,lsize,ldsc)
Search the HEPEVT common block for all the charged, stable descendants of particle ip. Fill ldsc(lsize).

ip = index of particle in HEPEVT
ndsc = number of particles found
lsize = length of the array
ldsc = array of particle indices in HEPEVT

stddautr(ip,ipc,lyn)
Check to see if particle ipc is a descendant of particle ip.

ip = index of particle in HEPEVT
ipc = index of prospective descendant in HEPEVT
lyn = true or false

stdrotboost(the,phi,bex,bey,bez)
Rotate and boost the frame of reference for particles in the HEPEVT common block.
This is a modified version of LUROBO from Jetset.

stddbrotb(imi,ima,the,phi,dbex,dbey,dbez)
Rotate and boost the frame of reference for the HEPEVT common block. Stddbrotb performs a specific range and double precision boost.
This is a modified version of LUDBRB from Jetset.

McFio

General Purpose i/o package.

Uses XDR to create a machine independent data representation.

Use RBIO to transfer events from memory to sequential media.

McFio features:

- Transportable data files
- Direct access for disk files
- Multiple i/o streams
- Support for various sequential media
- Version stamping for backward compatibility.
- Event structure

StdHep MCFio I/O Routines

stdxwrt(ilbl,istream,lok)
Write record type ilbl to an xdr output stream, returning lok.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxwinit(filename,title,ntries,istream,lok)
Initialize xdr and open an output stream, returning lok.
This routine should never be called more than once.

filename	=		name of the xdr output file
title	=		title for the xdr output file
ntries	=		expected number of events in this file
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxwopen(filename,title,ntries,istream,lok)
Open an xdr output stream, returning lok.
This routine is called by stdxwinit.

filename	=		name of the xdr output file
title	=		title for the xdr output file
ntries	=		expected number of events in this file
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

StdHep MCFio I/O Routines

stdxrd(ilbl,istream,lok)

Read the next record from an xdr input stream, returning ilbl and lok.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdxrdm(ilbl,istream,lok)

Read the next record from an xdr input stream, returning ilbl and lok.
stdxrdm handles multiple input streams to fake multiple interactions.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer
lok	=	0	if no problems were encountered

stdzero

Set the contents of the HEPEVT common block to zero.
This routine must be called by the user if stdxrdm is used.

StdHep MCFio I/O Routines

stdxrinit(filename,ntries,istream,lok)

Initialize xdr and open an input stream, returning ntries and lok.

This routine should never be called more than once.

filename	=	name of the xdr input file
ntries	=	expected number of events in this file
istream	=	index to array ixdrstr holding stream pointer
lok	= 0	if no problems were encountered

stdxropen(filename,ntries,istream,lok)

Open an xdr input stream, returning ntries and lok.

This routine is called by stdxrinit.

filename	=	name of the xdr input file
ntries	=	expected number of events in this file
istream	=	index to array ixdrstr holding stream pointer
lok	= 0	if no problems were encountered

stdxend(istream)

Close the xdr output stream.

istream = index to array ixdrstr holding stream pointer

StdHep MCFio I/O Routines – C Binding

int StdHepXdrRead(ilbl,istream)

Read the next record from an xdr input stream, returning ilbl.

The return argument is 0 if no problems were encountered.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer

int StdHepXdrReadMulti(ilbl,istream)

Read the next record from an xdr input stream, returning ilbl.

StdHepXdrReadMulti handles multiple input streams to fake multiple interactions.

The return argument is 0 if no problems were encountered.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer

StdHepZero

Set the contents of the hepevt_ structure to zero.

Hepevt_ is a mirror of the HEPEVT common block.

This routine must be called by the user if StdHepXdrReadMulti is used.

StdHep MCFio I/O Routines – C Binding

int **StdHepXdrReadInit**(filename,ntries,istream)
Initialize xdr and open an input stream, returning ntries.
The return argument is 0 if no problems were encountered.
This routine should never be called more than once.

filename = name of the xdr input file
ntries = expected number of events in this file
istream = index to array ixdrstr holding stream pointer

int **StdHepXdrReadOpen**(filename,ntries,istream)
Open an xdr input stream, returning ntries.
This routine is called by stdxrinit.
The return argument is 0 if no problems were encountered.

filename = name of the xdr input file
ntries = expected number of events in this file
istream = index to array ixdrstr holding stream pointer

StdHepXdrEnd(istream)
Close the xdr output stream at ixdrstr(istream).

istream = index to array ixdrstr holding stream pointer

StdHep MCFio I/O Routines – C Binding

int **StdHepXdrWrite**(ilbl,istream)
Write record type ilbl to an xdr output stream.
The return argument is 0 if no problems were encountered.

ilbl	=	1	HEPEVT common block
ilbl	=	2	HEPEVT and HEPEV2 common blocks
ilbl	=	100	StdHep begin run record
ilbl	=	200	StdHep end run record
istream	=		index to array ixdrstr holding stream pointer

int **StdHepXdrWriteInit**(filename,title,ntries,istream)
Initialize xdr and open an output stream.
The return argument is 0 if no problems were encountered.
This routine should never be called more than once.

filename	=	name of the xdr output file
title	=	title for the xdr output file
ntries	=	expected number of events in this file
istream	=	index to array ixdrstr holding stream pointer

int **StdHepXdrWriteOpen**(filename,title,ntries,istream)
Open an xdr output stream.
This routine is called by stdxwinit.
The return argument is 0 if no problems were encountered.

filename	=	name of the xdr output file
title	=	title for the xdr output file
ntries	=	expected number of events in this file
istream	=	index to array ixdrstr holding stream pointer

```
      program simple
C
#include 'ludat1.inc'
#include 'stdlun.inc'
      integer istr
      integer nevt,I,nprnt,lok
C
      nevt=100
      nprnt=1
C initialize StdHep logical units
      lnhwrt=23
      lnhrd=0
      lnhdcy=0
      lnhout=22
      mstu(11)=lnhout
      open(unit=lnhout,file='stdtstjx.lpt',status='NEW')
      call stdxwinit('stdtstjx.io','StdHep/Jetset example',
     1               nevt,istr,lok)
      call mcfio_FPrintDictionary(lnhout)
C           Write begin-run record
      call stdxwrt(100,istr,lok)
      IF(lok.NE.0) write(lnhout,*) ' begin run problem'
C event loop
      do 100 I=1,nevt
        call lu2ent(0.,2,-2,20.)
        call lunhep(1)
        IF(I.le.nprnt) call heplst(1)
        call analysis_routine
        call stdxwrt(1,istr,lok)
 100  continue
C           Write end-of-run record
      call stdxwrt(200,istr,lok)
      IF(lok.NE.0) write(lnhout,*) ' end run problem'
c...close event file
      call stdxend(istr)
c...close output file
      close(unit=lnhout)
      stop
      end
```

StdHep Event Display

A motif–based event display has been written by Paul Lebrun, using the Nirvana spin widget.

Space

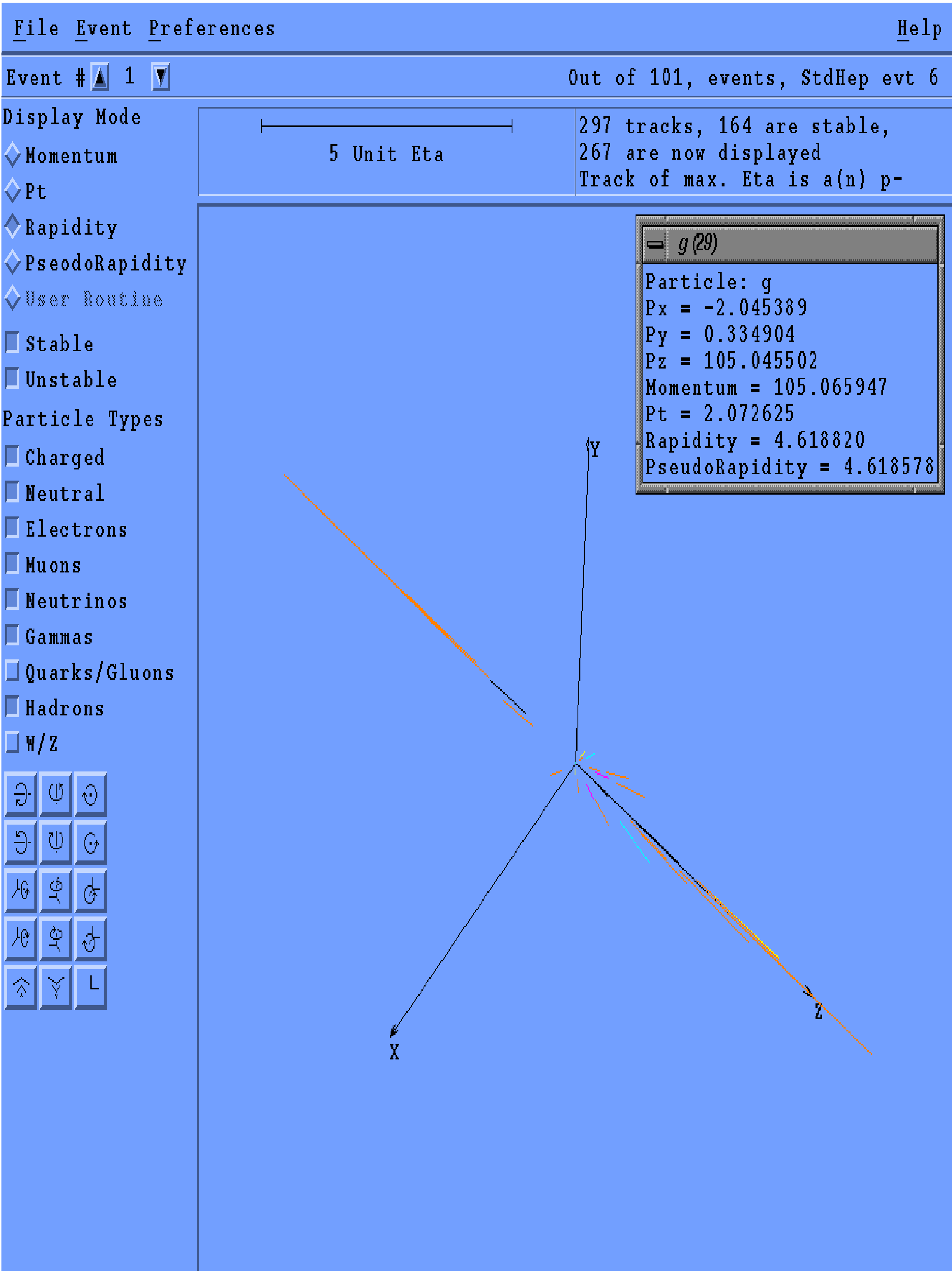
- **select, display, and browse tracks and vertices from the HEPEVT common block**
- **kinematics and spatial relationships**
- **decay chain.**

Phase

- **select, display, and browse StdHep events**
- **kinematics and topology**
- **decay chain.**

Phase can be invoked from Space, but Space cannot be invoked from Phase.

All the information in the HEPEVT common block can be accessed by interacting with this display.


File Event Preferences
Help
Event # 1
Out of 101, events, StdHep evt 6
Display Mode
Momentum
Pt
Rapidity
PseudoRapidity
User Routine
Stable
Unstable
Particle Types
Charged
Neutral
Electrons
Muons
Neutrinos
Gammas
Quarks/Gluons
Hadrons
W/Z
5 Unit Eta
297 tracks, 164 are stable,
267 are now displayed
Track of max. Eta is a(n) p-
g (29)
Particle: g
Px = -2.045389
Py = 0.334904
Pz = 105.045502
Momentum = 105.065947
Pt = 2.072625
Rapidity = 4.618820
PseudoRapidity = 4.618578
Y
X
Z

McFast

Parameterized Fast Monte Carlo System

Provide physics community with a tool that gives:

> the ability to compare acceptance for different detector configurations and detector subsystem design with minimal development work.
>
> a consistent format to compare different physics generators.
>
> the ability to study physics analysis cuts for signal and background for arbitrary processes via user hooks.
>
> the ability to visualize the detector configurations, detector subsystems and physics quantities to verify and elucidate the results.

McFast Framework

Written in FORTRAN 77 with some C.

Extensive use of FORTRAN structures.

Processing is controlled by a command procedure file.

Read in STDHEP events or internally generate particles.

Capability to overlay events to simulate multiple interactions.

Event output via McFio

User Control

Histogram booking and filling.

Event filtering.

Calls to simulation routines.

- → hit generation
- → tracing
- → parameterized track generation
- → calorimeter showering
- → muon tracking

Callback routine for use with command line input.

Flexible examples and defaults user routines supplied.

User tailors McFast with the user control routines.

Geometry Definition

The user specifies the geometry definition.

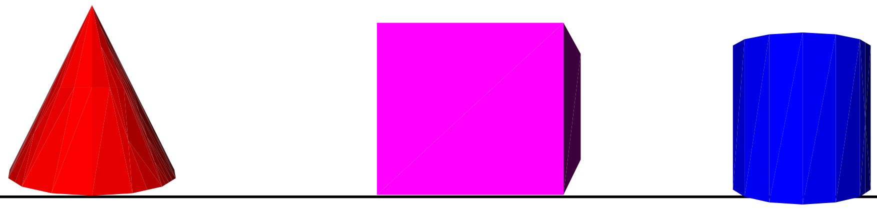
The structures and geometry code are created using DBin.

Updates or changes to the geometry can be read in at run time.

Elements and Devices:

Materials and mixtures
Magnets and magnetic fields
Beam pipes
Silicon detectors (barrels, disks, x–y planes)
Pixel detectors
Drift chambers
(Cerenkov Counters)
Muon detectors
Electromagnetic and hadronic calorimetry
Absorbers.

Elements are simple shapes:



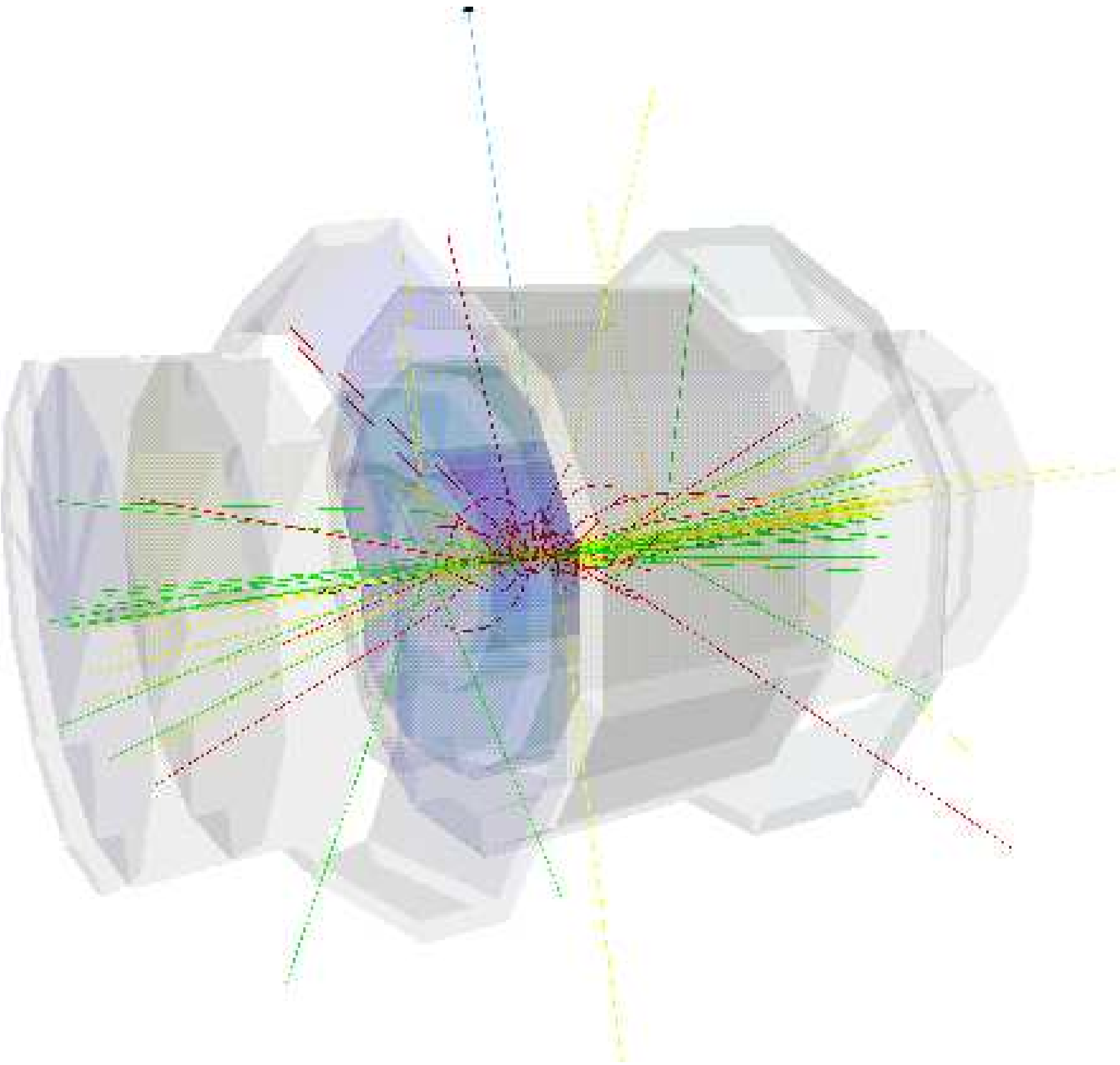
```
database mcfast 0100

include db/detector.db

make detector "CDF RUN2" "COMBINED"

! ************ Materials ***************

include db/materials.db

! ************ Mixtures ***************

include db/mixtures.db

! ************** Beam pipe ***************

include db/beampipe.db

make BPipe "beam" 0.00 2.00 0.00 1000. "VACU" 0.00   0.05 /
                    0.00  0.00  "VACU" "BE"  "VACU" "VACU"
```

```
! ************* Planar Silicon **************

include db/sixy.db

include db/sixy_layer.db

! **************** Barrel Silicon *****************

include db/sibarrel.db

make SiBarrel 1 "BSI "    8  –34.5000
make SiBarrel 2 "BSI "    8    0.0000
make SiBarrel 3 "BSI "    8   34.5000

include db/siblayers.db

make LayerSiB    1   1 "SILI"   12 34.0000  0.0000  2.4160
0.0300  1.3000   1
make LayerSiB    1   2 "SILI"   12 34.0000  0.0000  2.4460
0.0000  1.3000   2
make LayerSiB    1   3 "SILI"   12 34.0000  0.0000  3.4910
0.0300  1.8800   1

make LayerSiB    2   1 "SILI"   12 34.0000  0.0000  2.4160
0.0300  1.3000   1
make LayerSiB    2   2 "SILI"   12 34.0000  0.0000  2.4460
0.0000  1.3000   2
make LayerSiB    2   3 "SILI"   12 34.0000  0.0000  3.4910
0.0300  1.8800   1

include db/wafer.db

make Wafer "ALL" 1   1  0 0.0   216   –0.6500 –51.5000
0.0060   0.0   0.9000   0.0017   0.0  0.0
make Wafer "ALL" 1   2  0 0.0  2048   –0.6500 –17.0000
0.0166   0.0   0.9000   0.0048   0.0  0.0
make Wafer "ALL" 3   8  0 0.0  2048   –2.0000 –17.0000
0.0166   0.0   0.9000   0.0048   0.0  0.0
```

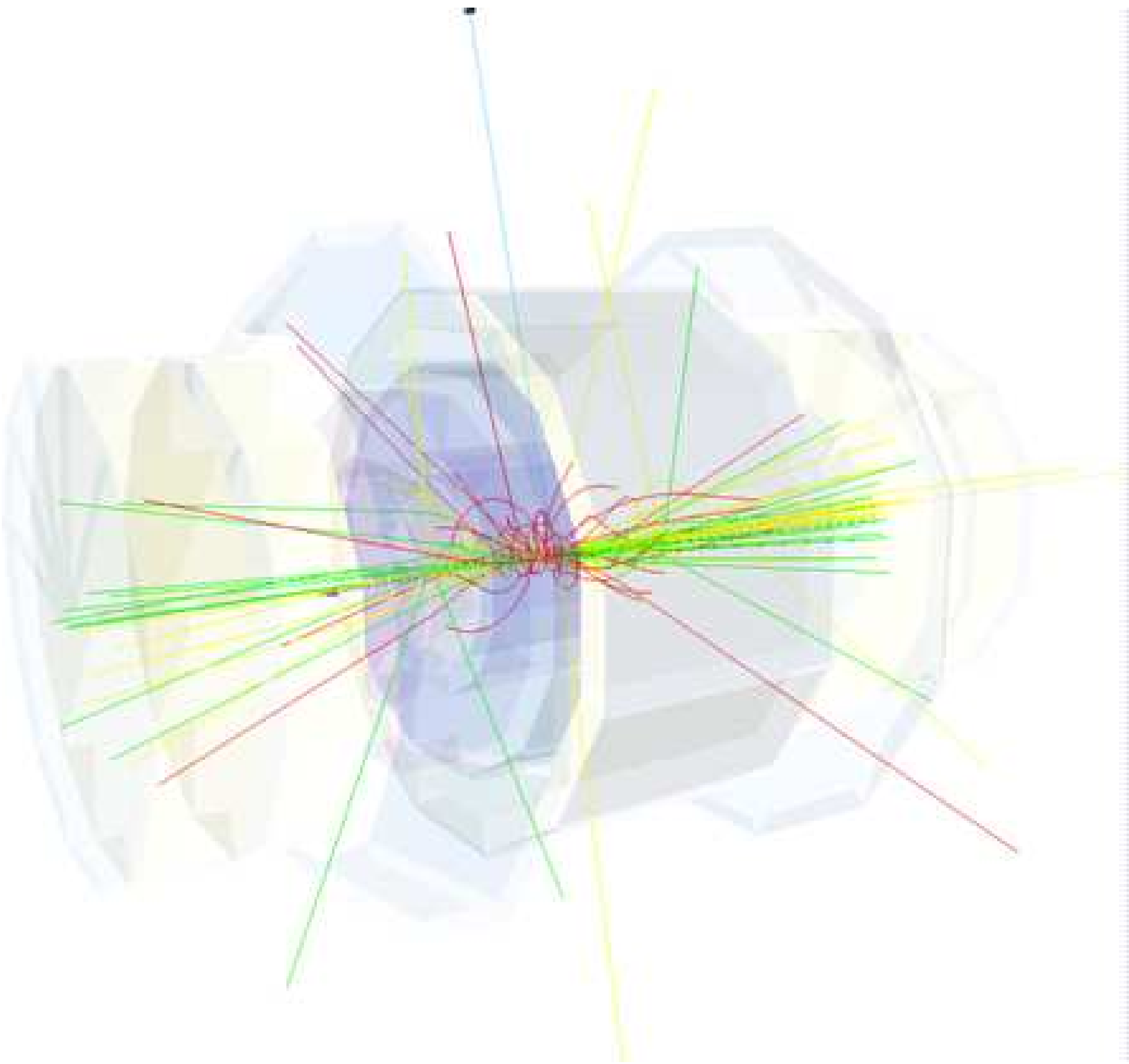
```
! ****************** Drift chamber *******************

include db/drift.db

make Drift 1 "DRFT" 84   0  25.0000  141.6050    0.0000 /
        307.4000 "AIR" 0.2030    0.0600    5.0800    5.0800 /
        "CFIB" "ALUM" "ALUM" "ALUM"

include db/drift_layer.db

make LayerDRFAno      1   1   30.9959  321.4000    0.6200  /
          30   1   0  –0.1258026E+00 0.0 0.0   0.9000    0.9600 /
          0.0200 0.0 0.0
make LayerDRFAno      1   2   31.6186  321.4000    0.6200  /
          30   1   0  –0.1008071E+00 0.0 0.0   0.9000    0.9600  /
          0.0200 0.0 0.0
make LayerDRFAno      1  84  132.0478  321.4000    0.6200 /
          120   1   0  0.2945637E–01 0.0 0.0   0.9000    0.9600 /
           0.0200 0.0 0.0

include db/drift_offset.db

make OffsetDRFAno  1   1  0.0   0.0  0.0   0.0   0.0  0.0   0.0 0.0
make OffsetDRFAno  1   2  0.0   0.0  0.0   0.0   0.0  0.0   0.0 0.0
make OffsetDRFAno  1 83  0.0   0.0  0.0   0.0   0.0  0.0   0.0 0.0
make OffsetDRFAno  1 84  0.0   0.0  0.0   0.0   0.0  0.0   0.0 0.0

include db/drift_cathode.db

! ******************* Forward tracking *********************

include db/ftrk.db

include db/ftrk_layer.db
```

```
! ****************** EM Calorimeter *********************

include db/emcal.db

make EMCal "Bemc"  "TUBE"     1 167.6400  167.6400 /
         207.2600  207.2600  0.0000  498.0000 "ALUM" /
         "ALUM" 24  20 0.01000    0.12000
make EMCal "PEMB"  "CONE"     2  11.8300    9.1000 /
         141.6050  141.6050   –200.0300   52.0600 "ALUM" /
         "ALUM" 72  20 0.01000    0.12000
make EMCal "PEMF"  "CONE"     2   9.1000   11.8300 /
         141.6050  141.6050  200.0300   52.0600 "ALUM" /
          "ALUM" 72  20 0.01000    0.12000

! ******************* Hcal *********************

include db/hadcal.db

make HADCal  "BHAC"  "TUBE"     1 207.2600  207.2600 /
            345.0900  345.0900   0.0000  498.0000 "ALUM" /
            24  16 0.01000    0.85000
make HADCal  "WHAB"  "CONE"     2 210.8000  143.7600 /
            318.7700  318.7700  –307.0630  116.1260 "ALUM" /
            24   4 0.01000    0.85000
make HADCal  "WHAF"  "CONE"     2 143.7600  210.8000 /
            318.7700  318.7700  307.0625  116.1250 "ALUM" /
            24   4 0.01000    0.85000
make HADCal  "PHB1"  "CONE"     2  13.0500   11.8300 /
           141.6060  141.6060 –237.5300   22.9400 "ALUM" /
           72  20 0.01000    0.85000
make HADCal  "PHF1"  "CONE"     2  11.8300   13.0500 /
           141.6050  141.6050   237.5300   22.9400 "ALUM" /
           72  20 0.01000    0.85000
make HADCal  "PHF2"  "CONE"     2  13.0500   19.1300 /
           143.7600  210.8000  307.0625  116.1250 "ALUM" /
           72  17 0.01000    0.85000
```

```
! ******************** Muon system ********************

include db/muon.db

make MUON  "CMU "  "TUBE"  1 347.0000  347.0000 /
       358.5000  358.5000    0.0000  452.0000  "AIR"  4
make MUON  "CMX1"  "CONE"  2 422.4000  561.0000 /
       457.6000  596.2000 –444.4000  158.4000  "AIR"  8

! **************** Absorbers ******************

include db/absorber.db

make Absorber  "BAB1"  "TUBE"  2  17.6000   17.6000  /
       390.0000  390.0000 –602.9000   97.8000  "IRON"

! ******************** Dipoles ************************

include db/dipole.db

! *********************** Solenoids ************************

include db/solenoid.db

make Solenoid  "Sole" 1.50000   0.0000  149.1000    0.0000  /
      498.0000 "AIR" 0.0000    2.0000    0.0000    0.0000  /
      "––––"  "MXSO"  "––––"  "––––"

! ****************** Beam vertex **********************
include db/beamvrtx.db

make BeamVrtx 0.0  0.0  0.0  0.003  0.003  30.00

! ***************** Hits per track ******************
include db/hitsontrack.db

make HitsOnTrack  30  5  3

end
```



L. Garren Dec. 6, 1995



L. Garren Dec. 6, 1995

Particle Tracing

All stable particles are traced through the detector volume until they shower, decay, pair convert, are absorbed or reach the end of detector volume.

The particle path is governed by the equations of motion in a magnetic field.

Tracing stations occur at detector planes or radii, magnet boundaries...

The particle track parameters (four momentum + position) are stored at each intersection.

Trace positions are used for track visualization.

The trace information is used to generate idealized hits in the active regions of the detectors.

Parameterized Track Generation

Process traces track from outer most point to production point

Accounts for measurement error, multiple scattering and energy loss

At each interesection, effects of detector material and measurement error are added to covariance matrix.

Track parameters and covariance matrix are moved to next intersection.

Generated track parameters are smeared by the covariance matrix at the production points.

Stored as a track if user specified hit requirements are passed.

Muon Tracking

Tracks are extended to the muon system.

Hits in the muon detectors are created.

Energy loss is calculated.

If a hit is present in the muon system, the track is stored as a muon.

Vertexing

Particles constrained to motion in magnetic field.

Vertexing algorithm works in a solenoid, dipole or field–free region.

User selects tracks to fit to vertices.

Can build particle at the vertex.

Calorimetry

Electromagnetic calorimetry

Parameterized showering for electrons and photons.   MIP trace for muons.

Transverse and longitudinal shower profile is parameterzied.

Showering continues until 95% of energy has been deposited or shower leaves calorimeter volume.

Energy from shower is spread across $\eta-\phi$ divisions of calorimeter consistent with shower profile.

After all particles have showered, contributions to each $\eta-\phi$ cell is smeared according to the specified resolution.

Parameterized hadron calorimetry is being developed.

Version 2.0 Features

Version 2.0 declared in April 95

Forward geometries added to program.

– Forward elements include:

Rectangular silicon planes at constant z
Rectangular tracking chambers at constant z
Dipole magnets (uniform fields)
Absorber material

– Central elements include:

Solenoid magnet (uniform field)
Cylindrical drift chambers
Barrel silicon planes
Calorimeters (not yet active just material)
Absorber material
Muon chambers
Beampipe

– Have written geometry files to model CDF, D0, LHC–B (tracking only).

– Need to flesh out forward elements.

Vertex fitting package handles vertices in a solenoidal field region or a zero field region. Working on fitting in a dipole field.

3D Graphics display of detector done, still working on event display.

This version well suited to vertexing studies.

Near term improvements:

Central and forward elements in a single geometry file.

– Try hybrid detectors with a central solenoid and dipoles on the ends.

– Solves beampipe problem for forward geometry.

Vertex fitting in a dipole field.

– We are very interested in looking at a central dipole to get large acceptance.

Muon chambers for forward detectors.

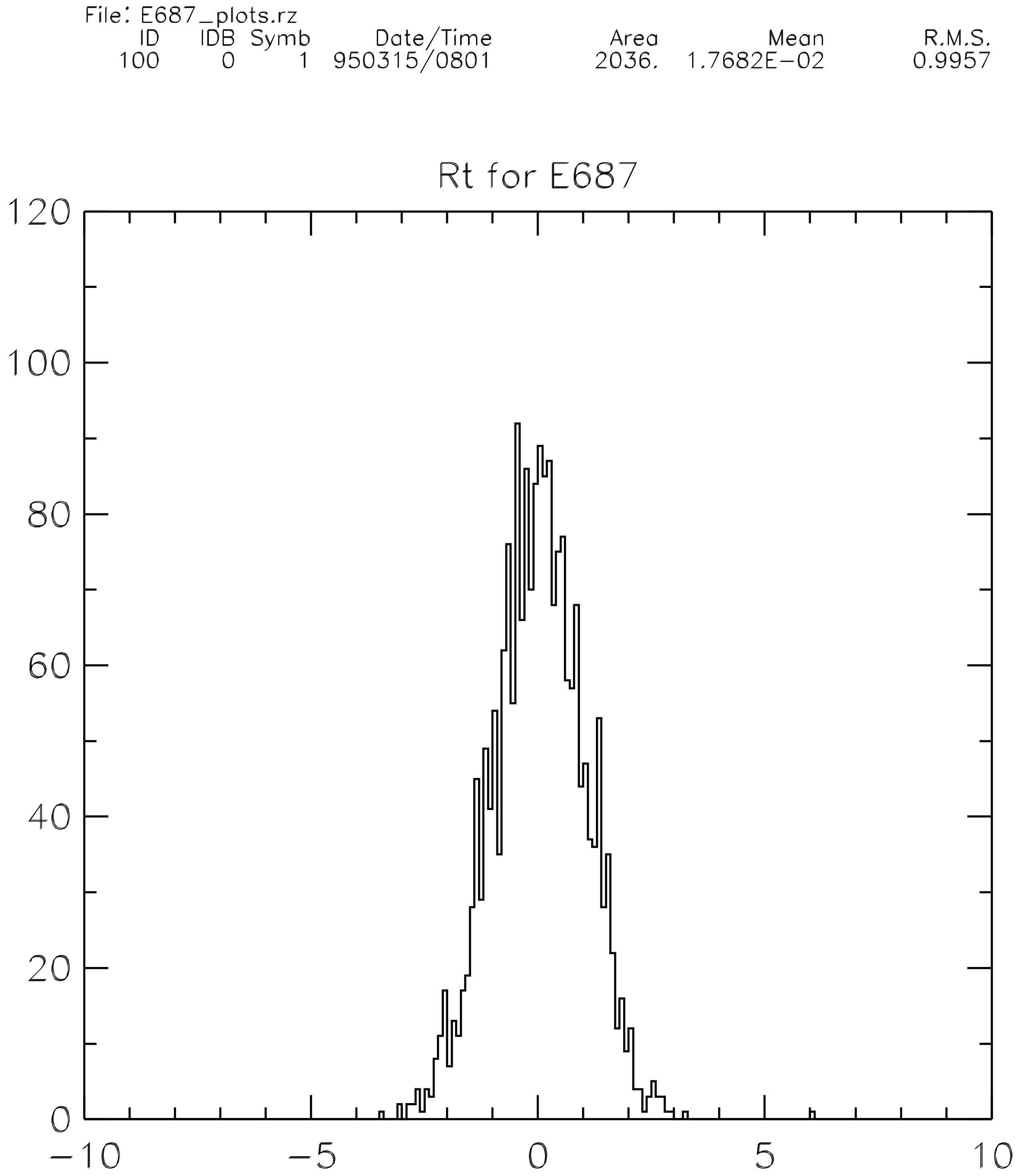
Ntuples in MCFio

Long term improvements:

can't write pattern recognition routines for each detector configuration

combine forward and central track confusion

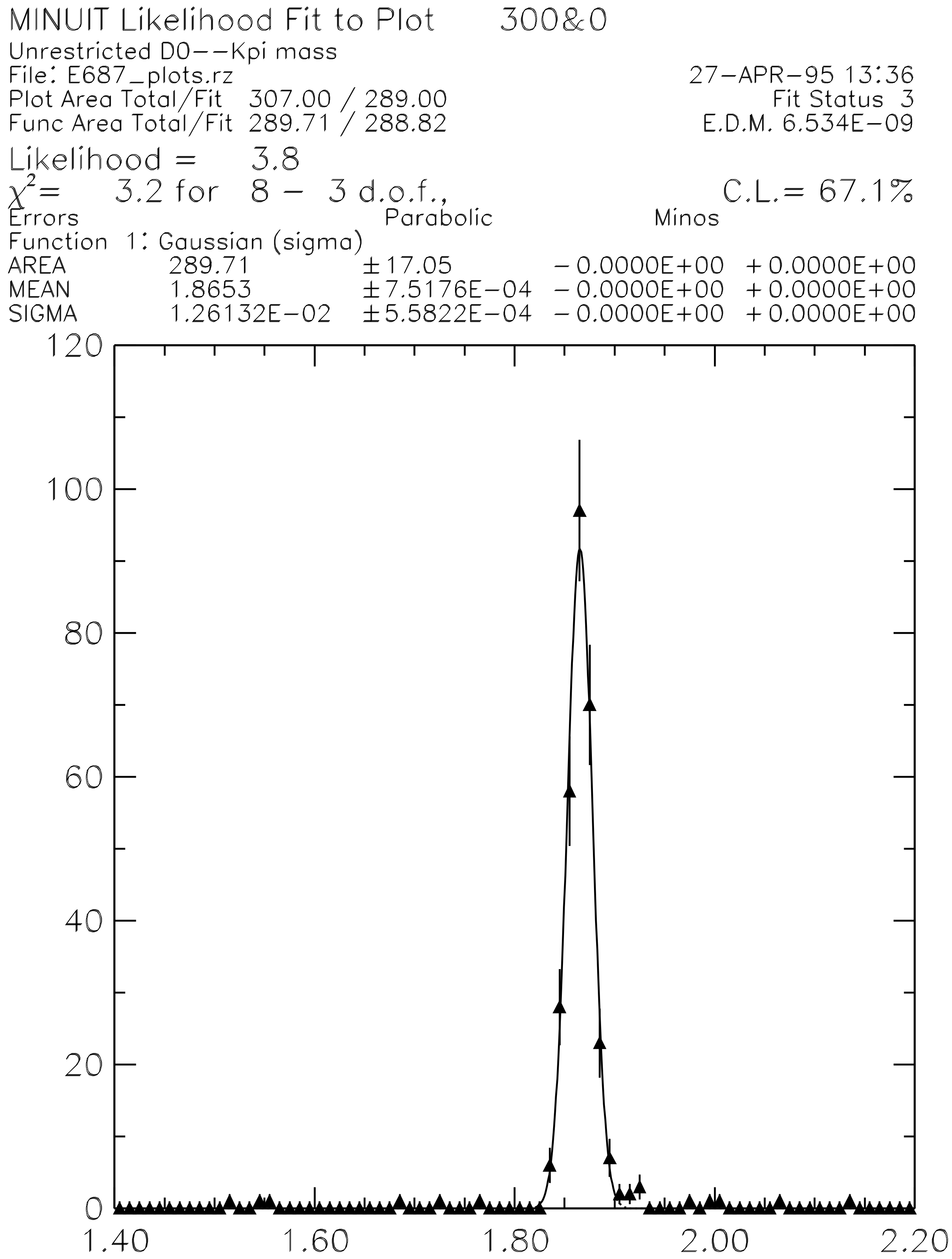
Modeling E687 Momentum Resolution



$R_t = (p_1 - p_2)/\sigma$

R_t should be centered at 0 and have $\sigma = 1$

E687 D^0 mass resolution



Validation of the Monte Carlo

Exact calculations

- Fixed target vertex detector
  - 30 GeV tracks
  - No multiple scattering

	σ_x μ	$\sigma_{\theta x}$ μr	σ_y μ	$\sigma_{\theta y}$ μr
exact	5.8	38	4.0	23
mcfast	5.6	37	3.9	23

  - with multiple scattering

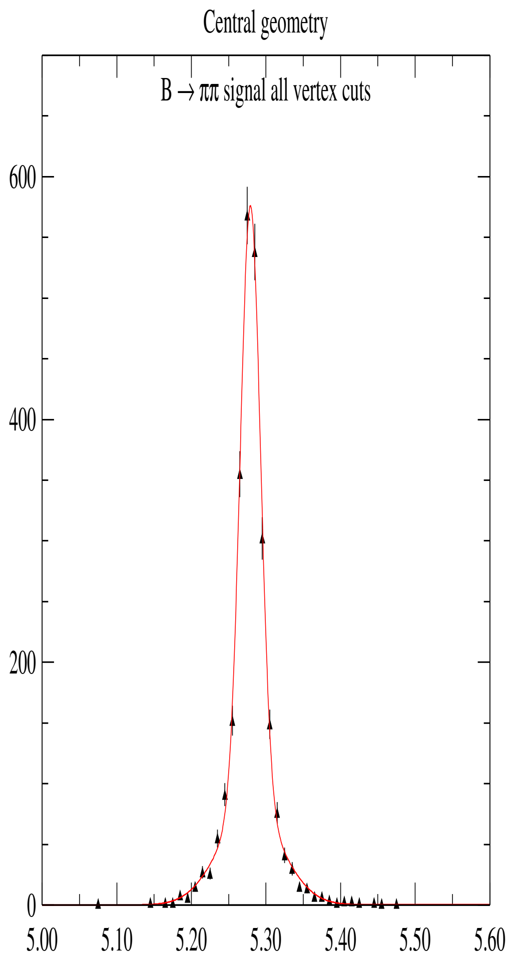
	σ_x μ	$\sigma_{\theta x}$ μr	σ_y μ	$\sigma_{\theta y}$ μr
exact	7.7	80	6.8	74
mcfast	7.5	77	6.6	72

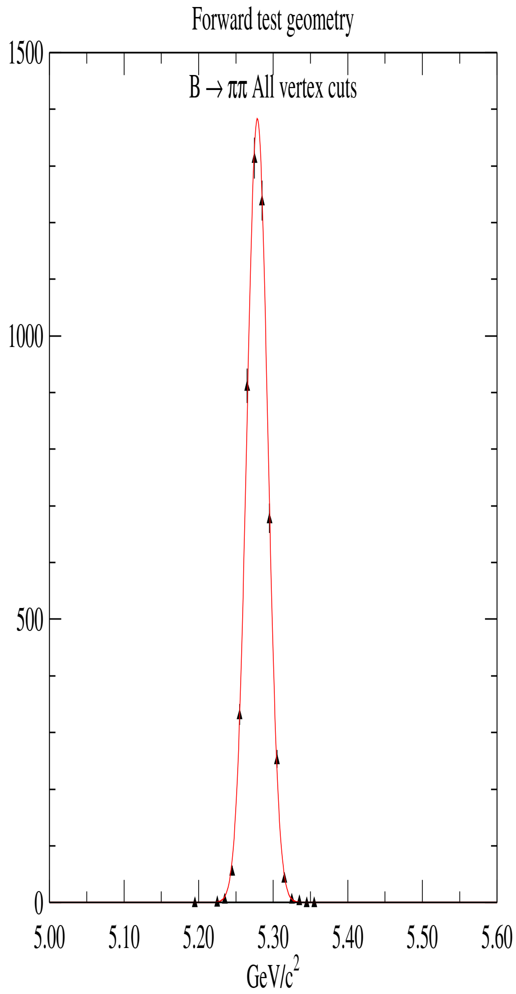
- Central detector did a element by element comparison of the covariance matrix between mcfast and an exact calculation with agreement at the 1–2% level

- Compare to E687 and D0

Timing and Verification

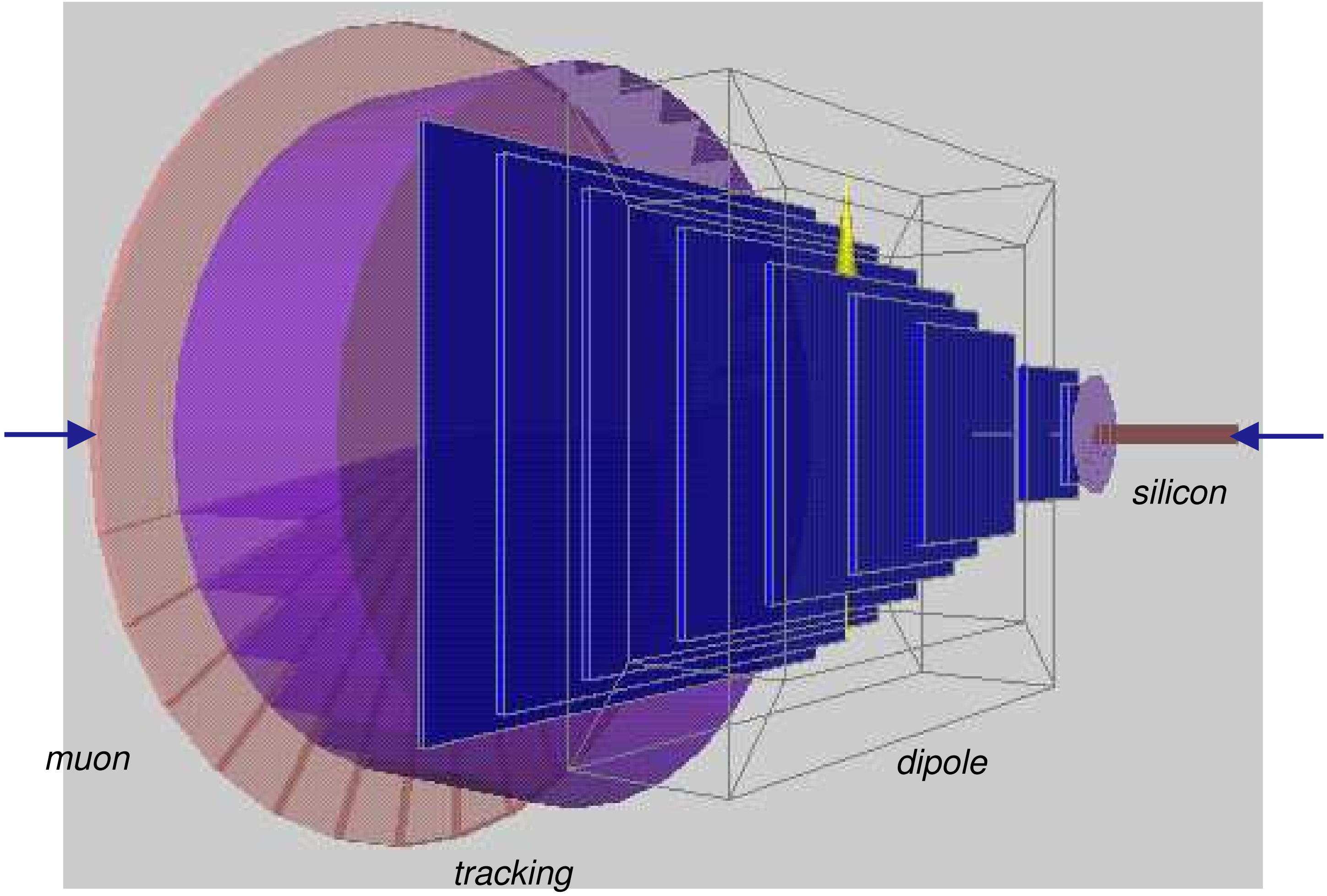
Timing: 0.4 cpu seconds/event on an R4400 Processor with 150 MHz clock for a CDF–like geometry for tracking.





B mass resolutions for two different geometries. Consistent with CDF.

Model of Test Forward Geometry



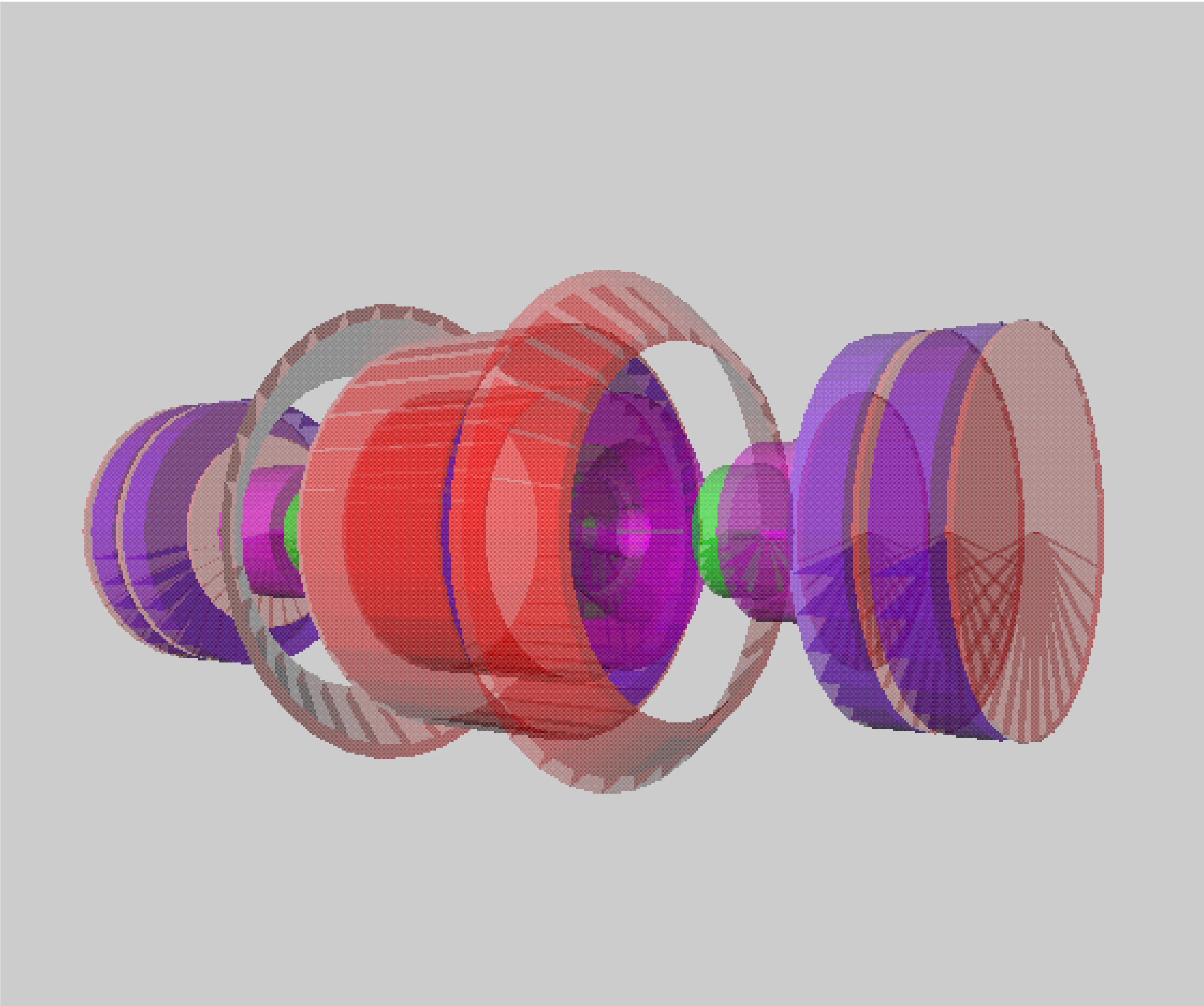
Large acceptance dipole magnet

28 plane Silicon vertex detector

Large tracking chambers

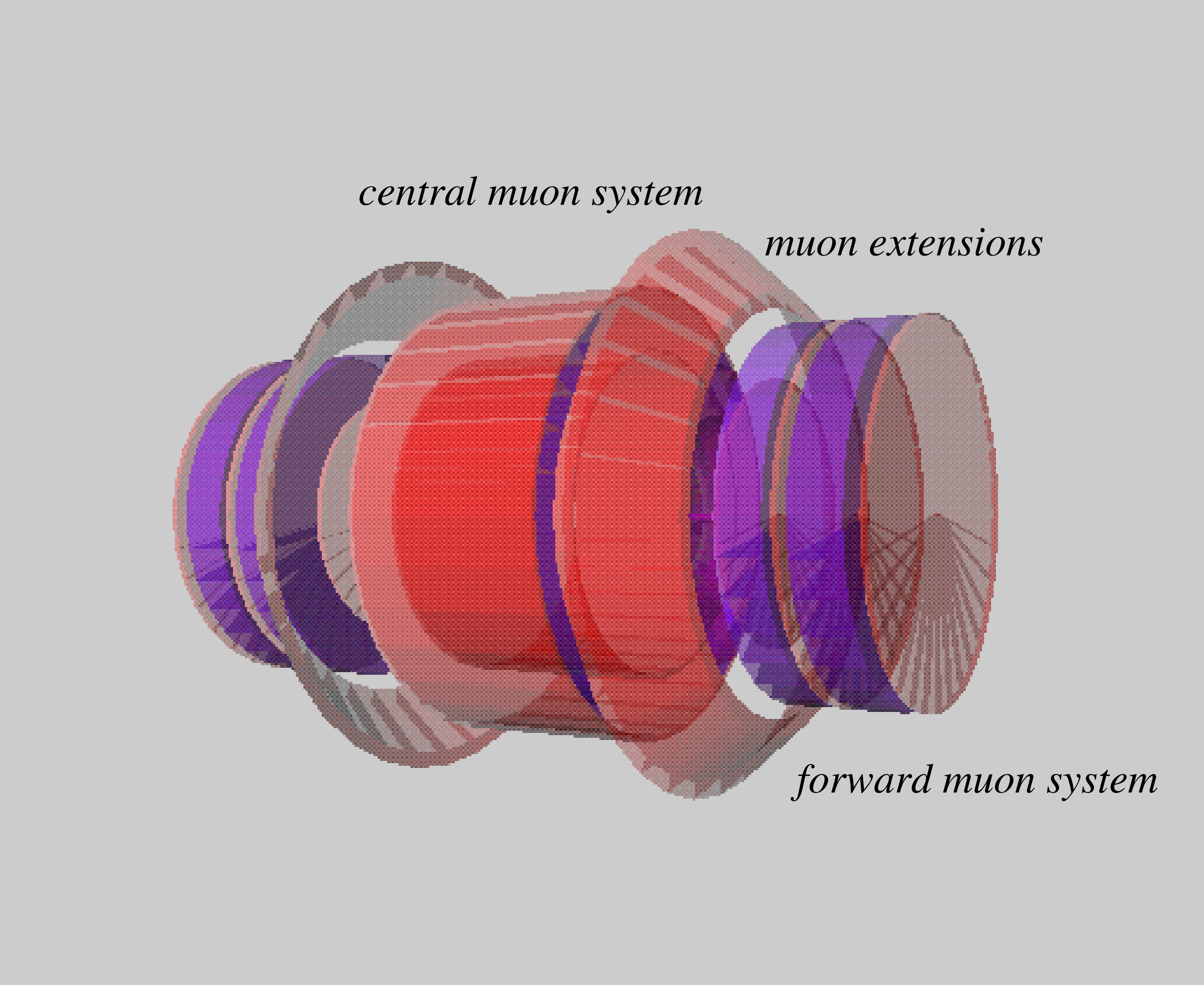
Muon identification

CDF – RUN I configuration



L. Garren Dec. 6, 1995

Model of CDF–like central detector



Modeled on Run II CDF design:
Solenoid
central drift chamber
silicon vertex detector
muon system

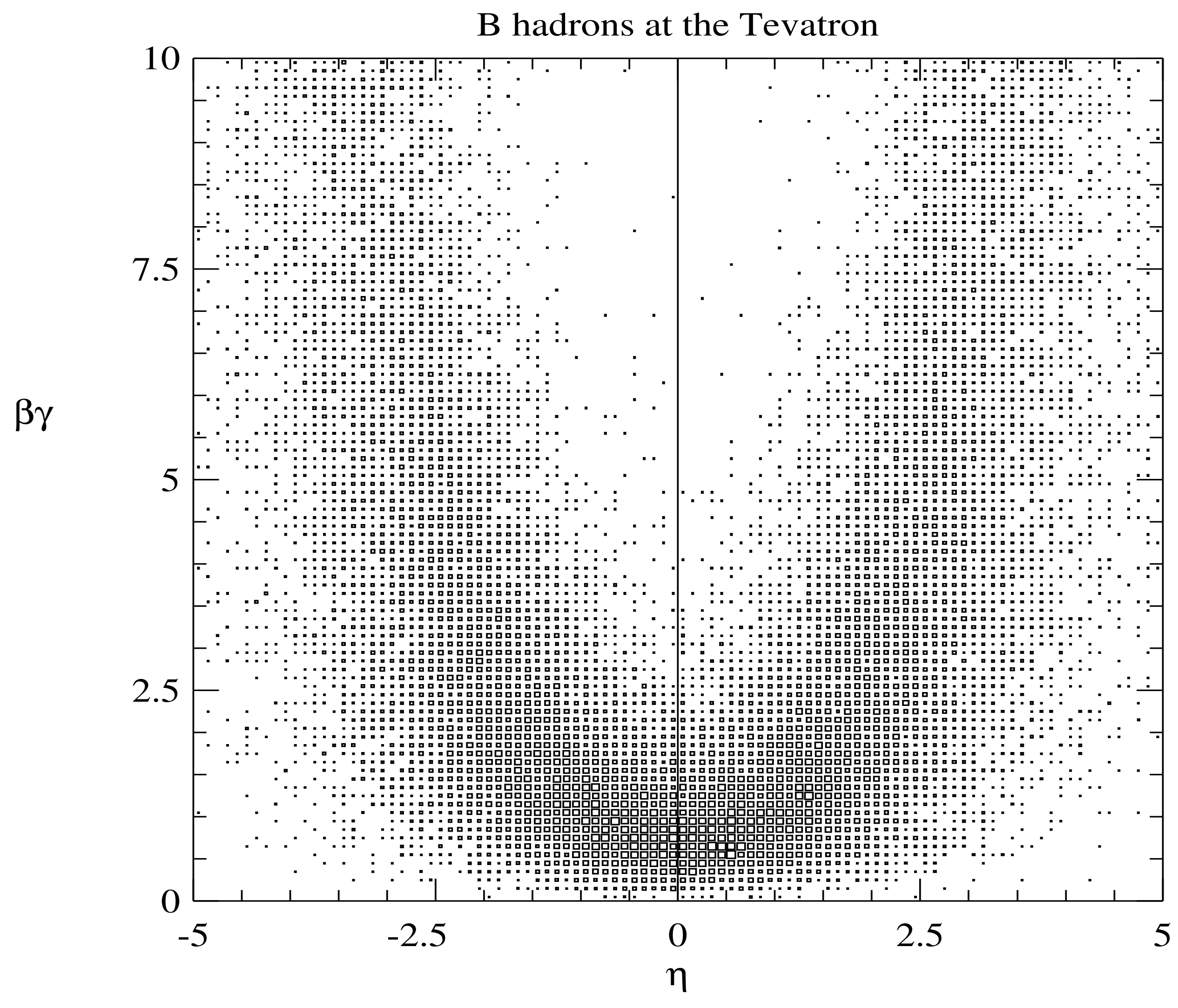
No scintillating fiber tracking yet
Calorimetry code under development

In designing a detector for B physics one optimizes acceptance against background rejection.

B cross section is larger at small $|\eta|$. **(Central)**

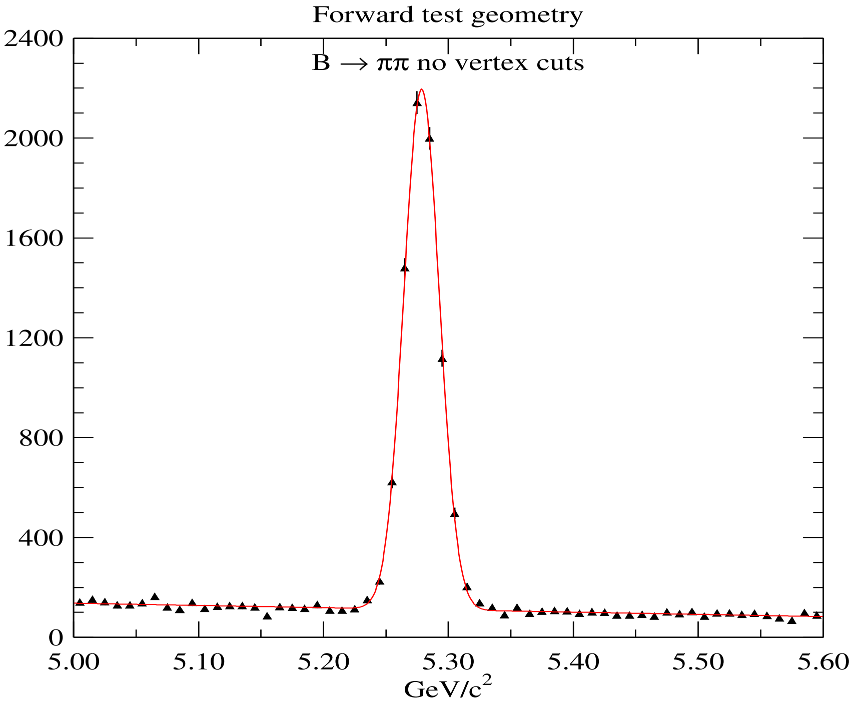
B decay distances are larger at large $|\eta|$. **(Forward)**

$\beta\gamma$



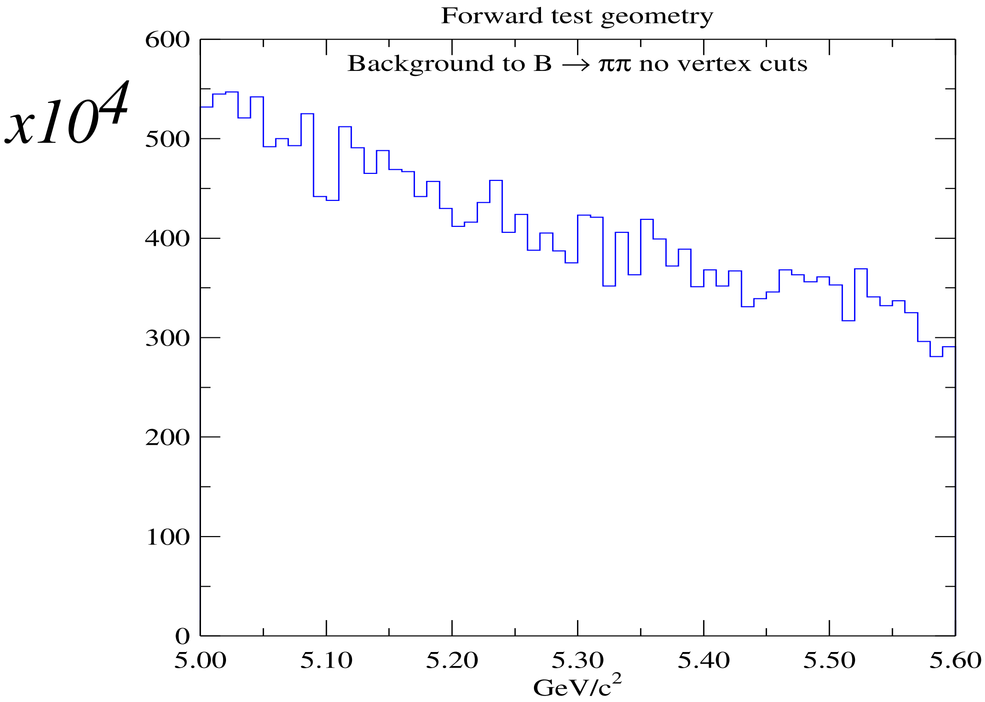
η

Forward Detector **No Vertex cuts**



$B^0_d \rightarrow \pi^+ \pi^-$

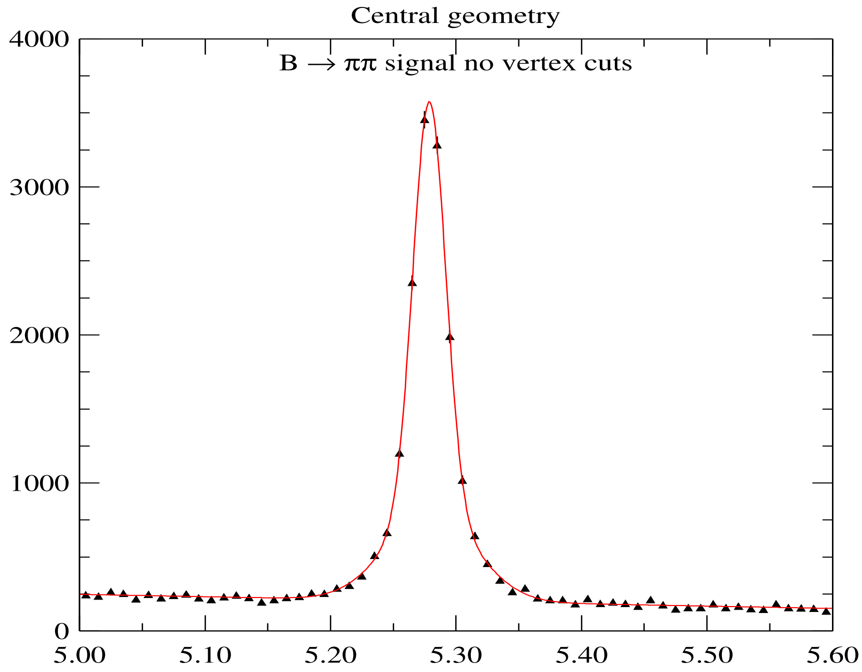
$\varepsilon = 18.6\%$



bb events

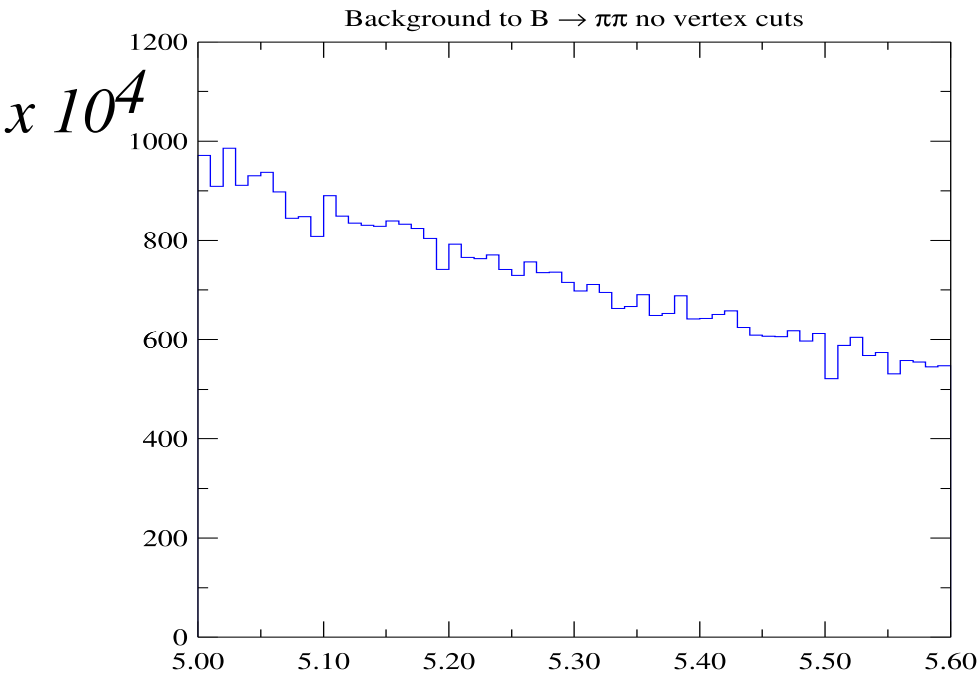
$\varepsilon = 1.2\%$

Central Detector **No Vertex cuts**



$B^0_d \rightarrow \pi^+ \pi^-$

$\varepsilon = 35.5\%$

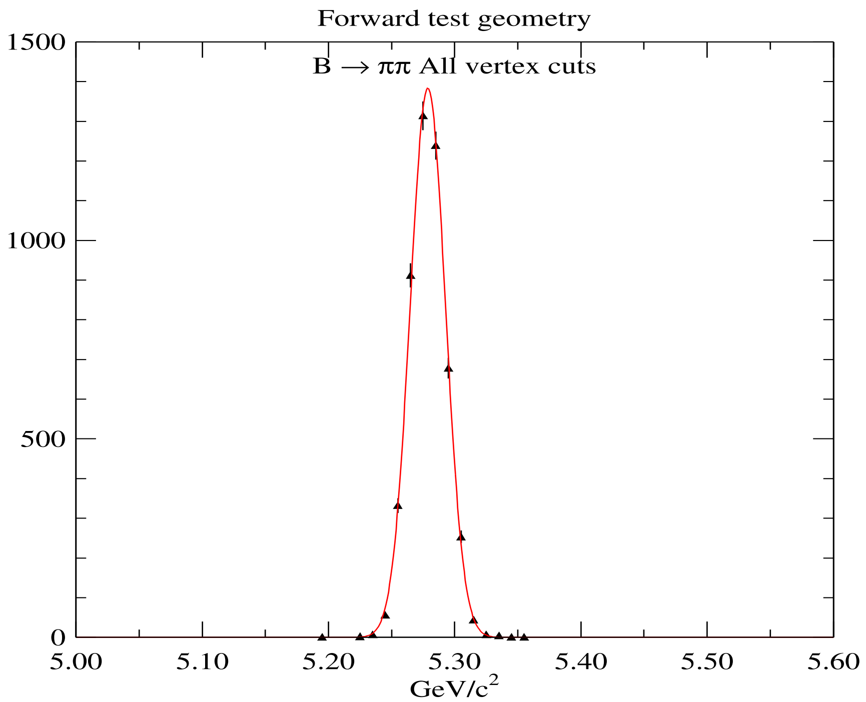


bb events

$\varepsilon = 2.2\ \%$

Forward Detector

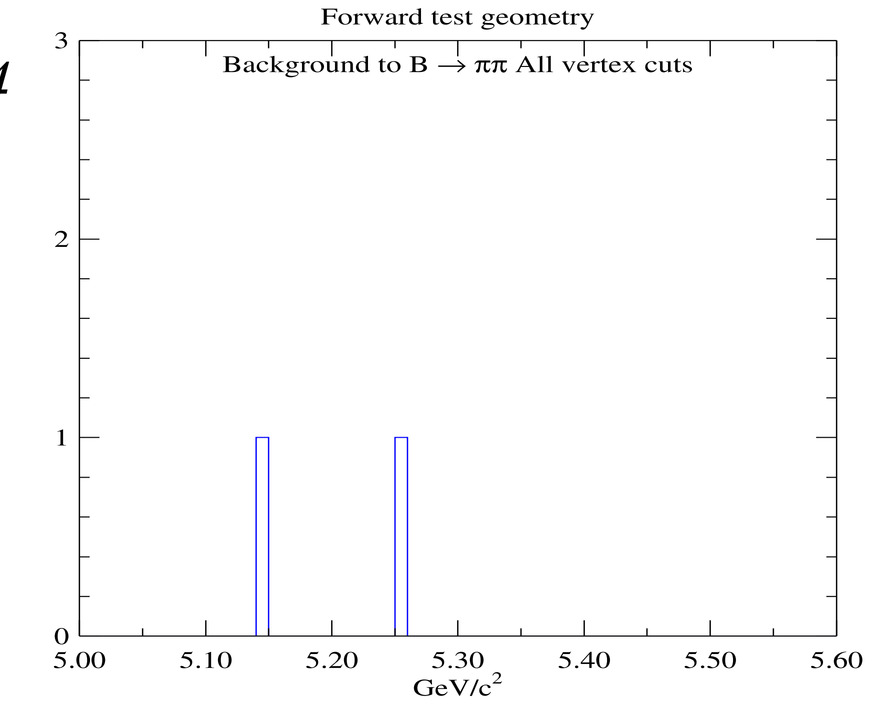
All vertex cuts



$B^0_d \rightarrow \pi^+ \pi^-$

$\varepsilon = 12.1\%$

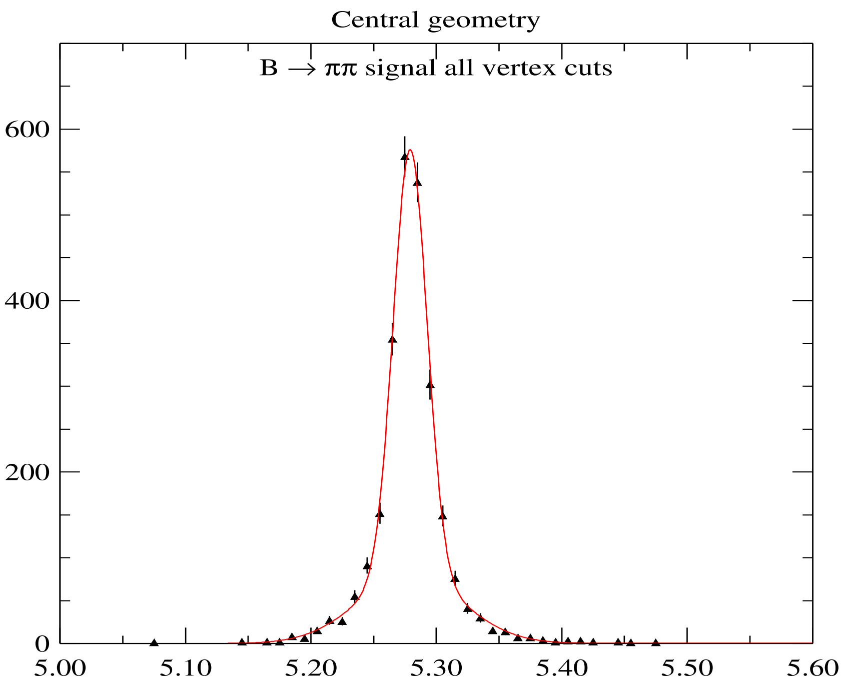
x10^4



bb events

$\varepsilon = 10^{-6}$

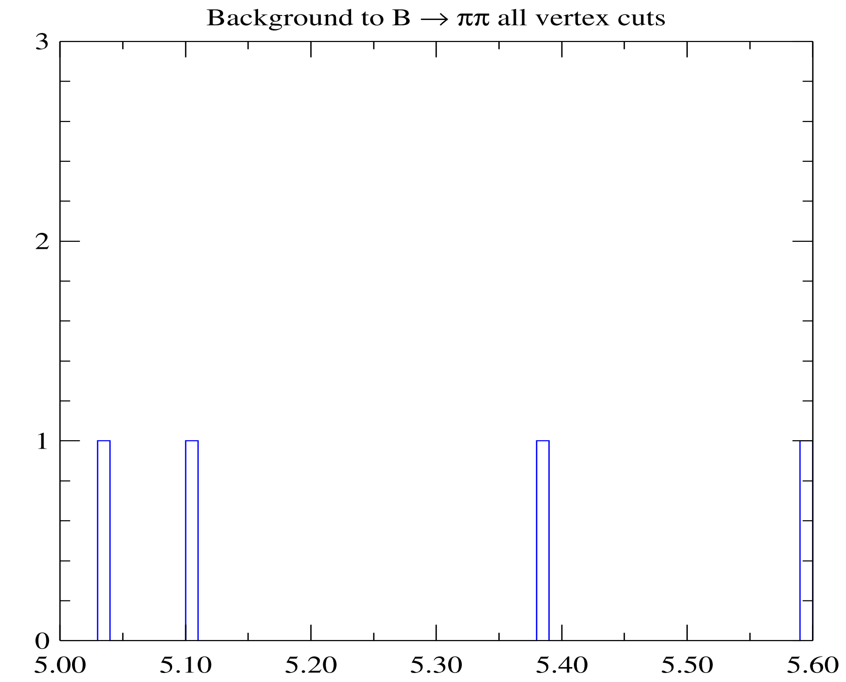
Central Detector **All vertex cuts**



$B^0_d \rightarrow \pi^+ \pi^-$

$\varepsilon = 6.2\%$

x10⁴



bb events

$\varepsilon = 2\mathrm{x}10^{-6}$

L. Garren **Dec. 6, 1995**

Further Information

http://fnpspa.fnal.gov/simulation.html

McFast:
http://fnpspa.fnal.gov/mcfast_docs.html

McFio:
http://fnpspa.fnal.gov/mcfio.html

DBin:
http://fnpspa.fnal.gov/tw–html/dbin.html

STDHEP:
http://fnpspa.fnal.gov/stdhep.html

Fitting Theory and Techniques:
http://www.phys.ufl.edu/avery/html/fitting.html