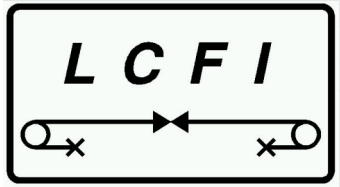


Vertex Tools for the ILC

Vancouver Linear Collider Workshop 2006

University of British Columbia

Joel Goldstein

CCLRC Rutherford Appleton Laboratory

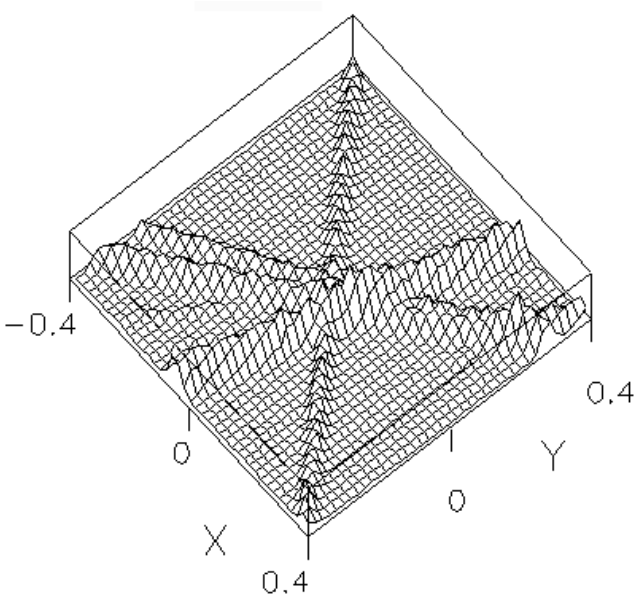
for the LCFI Collaboration

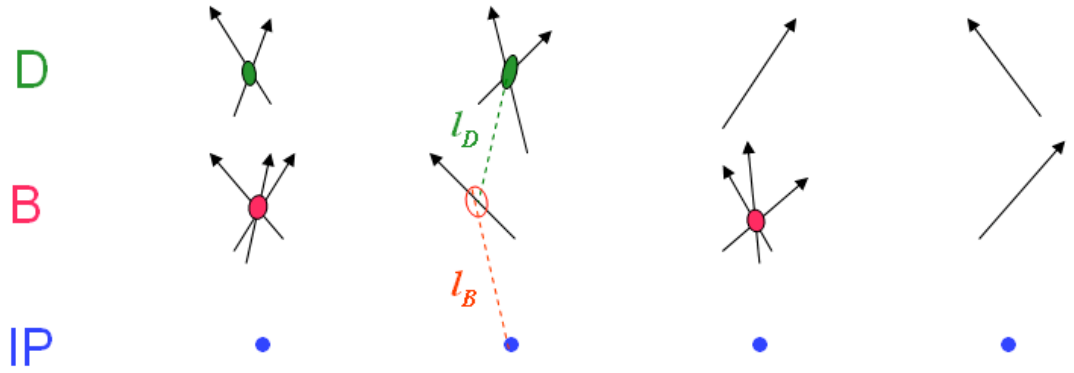
LCFI Vertex Package

- **Integrated vertexing in ILC software**
  - *still using SGV for studies*
- **Comprehensive package of tools**
  1. **Vertex finding & fitting**
  2. **Flavour tagging**
  3. **Vertex charge**
- **Interface to existing software**
  - **LCIO persistency framework**
    - *New Vertex Class?*
  - **C++ / MarlinReco**

Vertex Finders

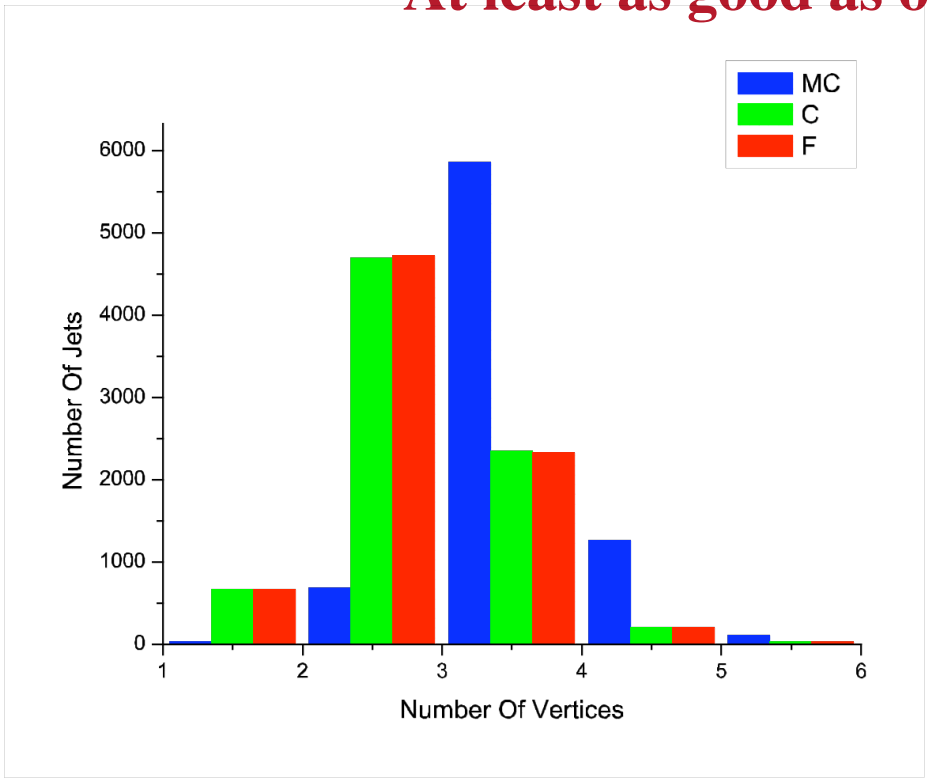
- **ZVTOP *(SLD)***
- **Two algorithms:**
  1. **ZVRES – topological finder**
  2. **ZVKIN – *"Ghostrack"***

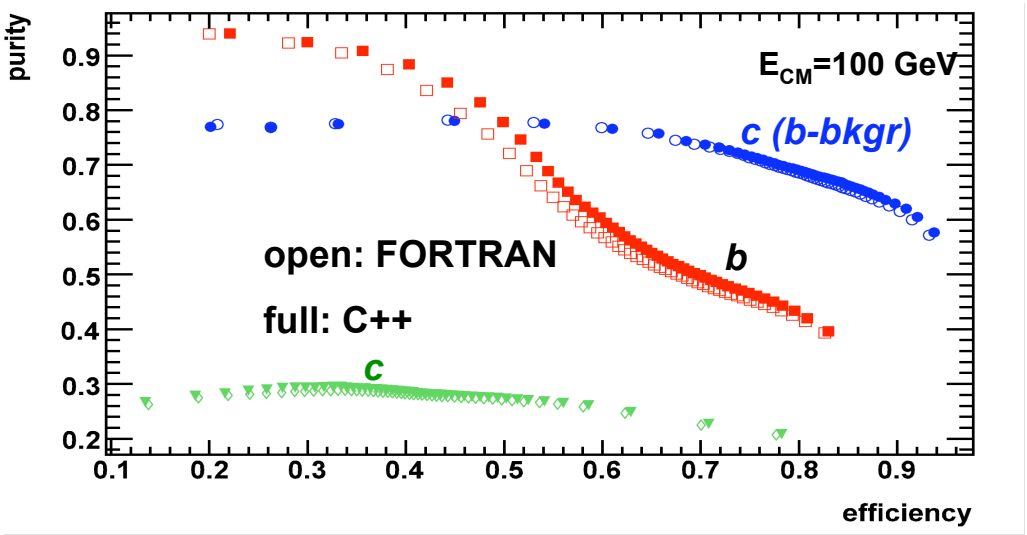




ZVTOP Validation

- **ZVRES checked against MC and FORTRAN**
  - **At least as good as old….**





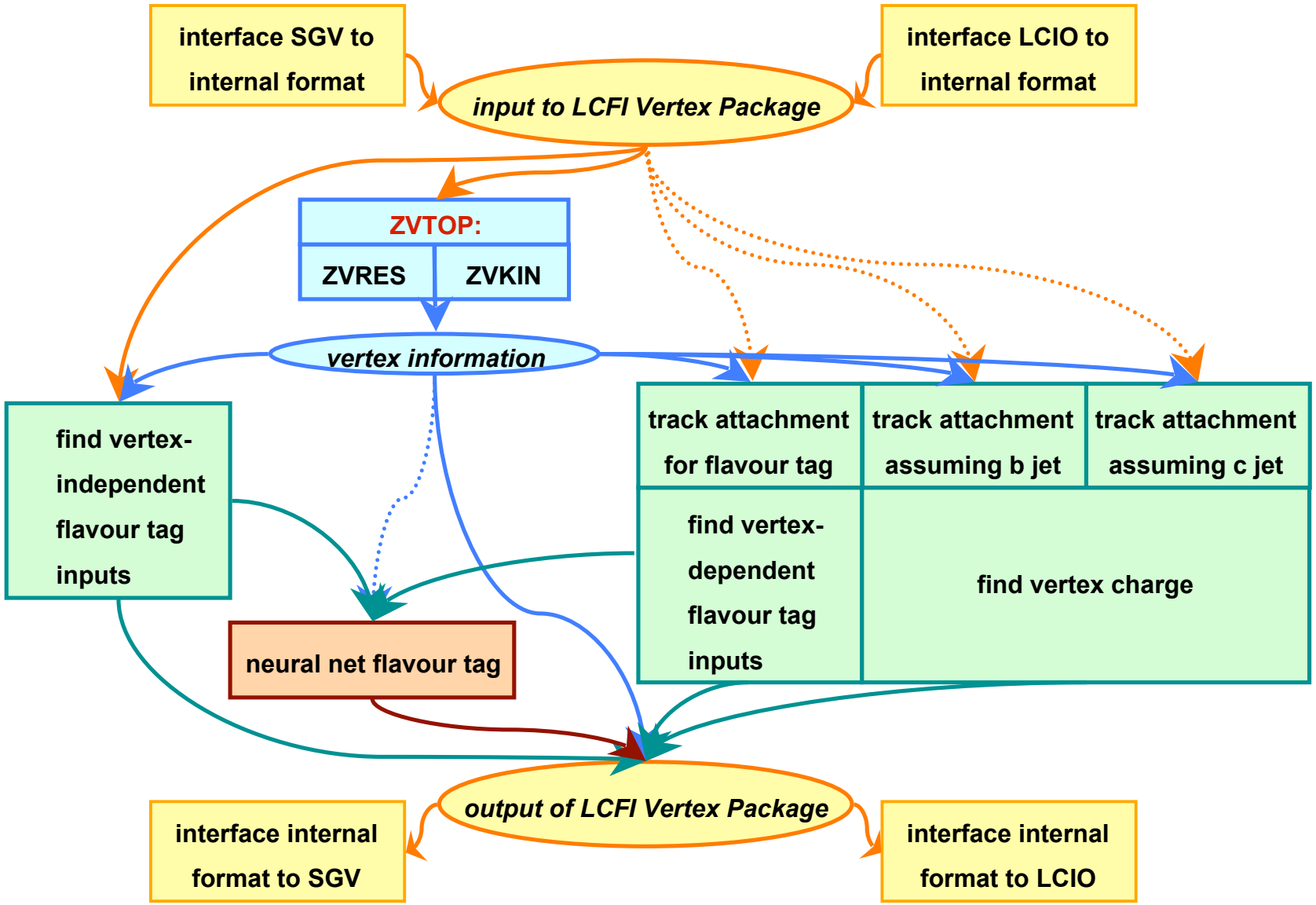
NB: non-realistic errors for testing

- **VZKIN coded, checks beginning**

Further Functionality

- **Flavour tag**
  - **Neural net based**
  - **Vertex, kinematic information as inputs**
- **Vertex Charge**
  - **Determine charge of b or c hadron**
- ***Future:***
  - ***Charge dipole***
  - **…**

Package Structure



Current Status

- **Vertex finding algorithms written**
  - ***Validation OK so far***
- **Flavour tagging/NN interface being written**
- **Infrastructure and interfaces under development**
- **Target:**

Release by end of Summer

Backup Slides

The ZVTOP vertex finder

D. Jackson,

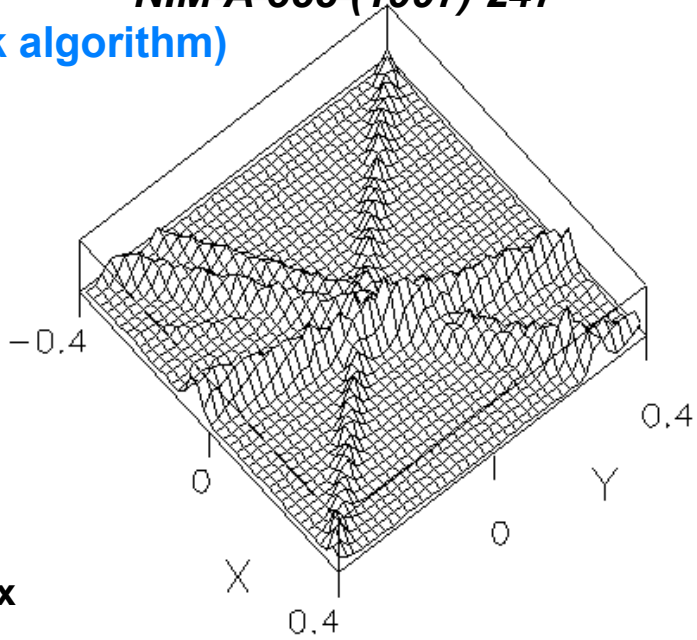
NIM A 388 (1997) 247

two branches: ZVRES and ZVKIN (also known as ghost track algorithm)

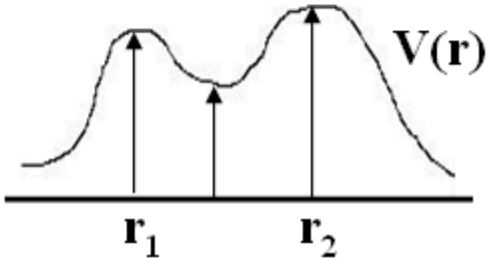
The ZVRES algorithm:

- **tracks approximated as Gaussian ´probability tubes´**
- **from these, a ´ve**

$$V(\mathbf{r}) = \sum_{i=0}^{N} f_i(\mathbf{r}) - \frac{\sum_{i=0}^{N} f_i^2(\mathbf{r})}{\sum_{i=0}^{N} f_i(\mathbf{r})}$$



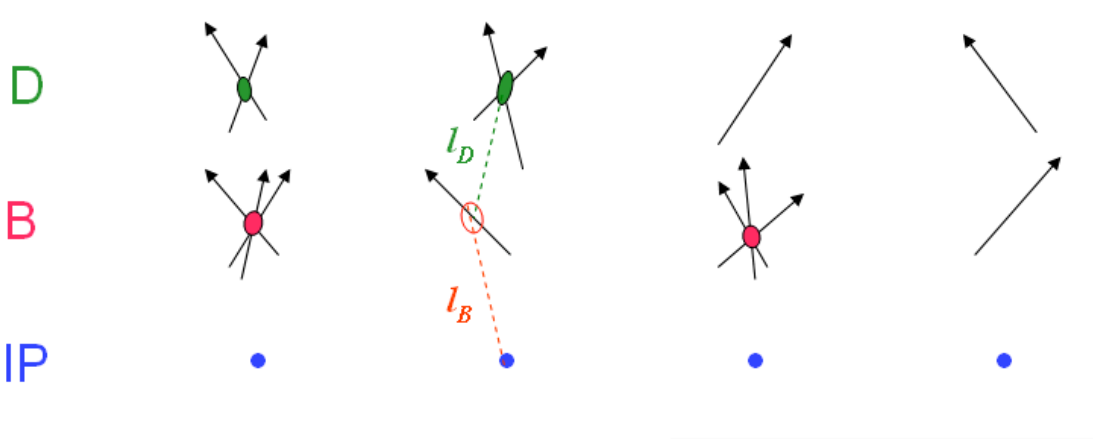
- **3D-space searched for maxima in the vertex function that satisfy resolubility criterion; track can be contained in > 1 candidate vertex**
- **iterative cuts on χ^2 of vertex fit and maximisation of vertex function results in unambiguous assignment of tracks to vertices**

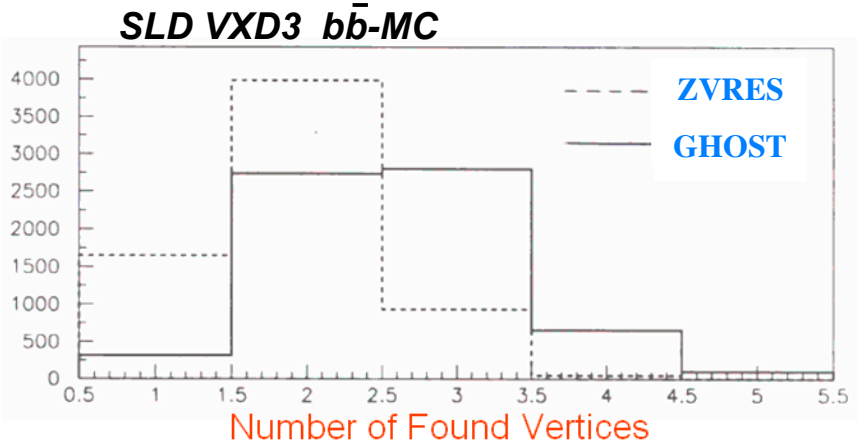


- **has been shown to work in various environments differing in energy range, detectors used and physics extracted**
- **very general algorithm that can cope with arbitrary multi-prong decay topologies**

The ZVKIN (ghost track) algorithm

- more specialised algorithm to extend coverage to b-jets in which one or both secondary and tertiary vertex are 1-pronged and / or in which the B is very short-lived;
- algorithm relies on the fact that IP, B- and D-decay vertex lie on an approximately straight line due to the boost of the B hadron





- should improve flavour tagging capabilities

Comparison of track-vertex association

- tables show percentages of tracks from the primary, B- and D-hadron decay at MC level that are assigned to primary, secondary (and tertiary) vertex or left isolated by ZVRES;

Mark Grimes

C++ ZVRES

Monte Carlo track origin	Two vertex case			Three vertex case			
	pri	sec	iso	Pri	sec	ter	iso
Primary	98.1	0.7	1.2	96.9	2.2	0.1	0.9
B decay	8.3	75.6	16.1	2.3	89.1	4.7	3.9
D decay	2.3	79.4	18.3	0.6	17.5	77.5	4.5

FORTRAN ZVRES

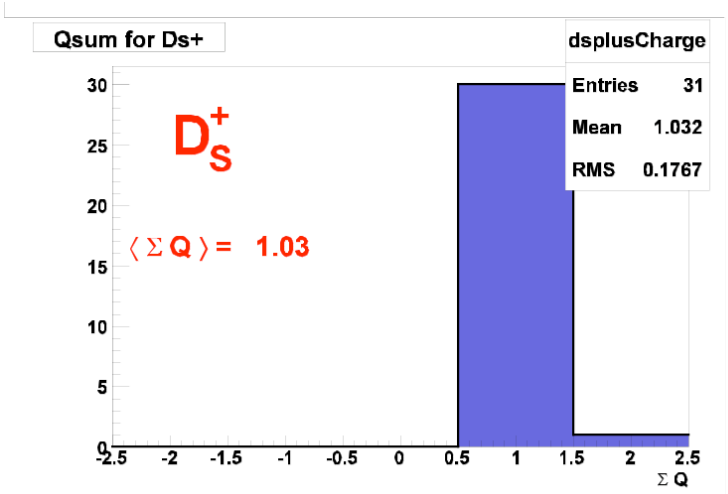
Monte Carlo track origin	Two vertex case			Three vertex case			
	pri	sec	iso	pri	sec	ter	iso
Primary	98.0	0.6	1.4	97.3	1.5	0.0	1.2
B decay	9.3	74.8	15.9	2.6	90.6	3.9	2.9
D decay	2.5	81.1	16.4	0.6	17.2	79.0	3.2

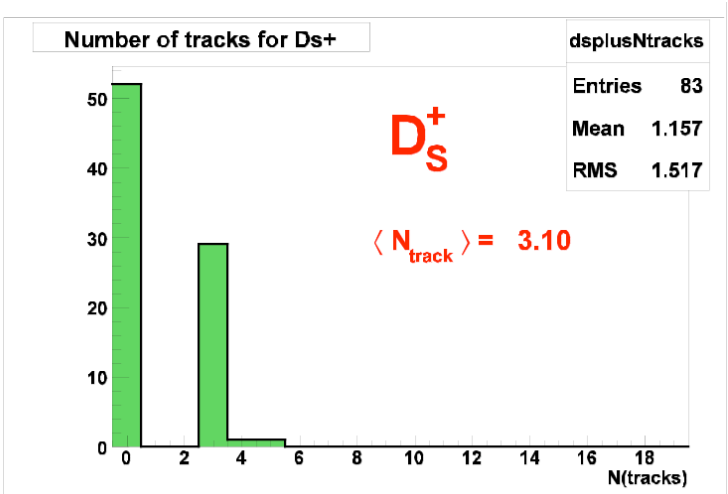
- excellent agreement also seen in this detailed comparison of the two versions

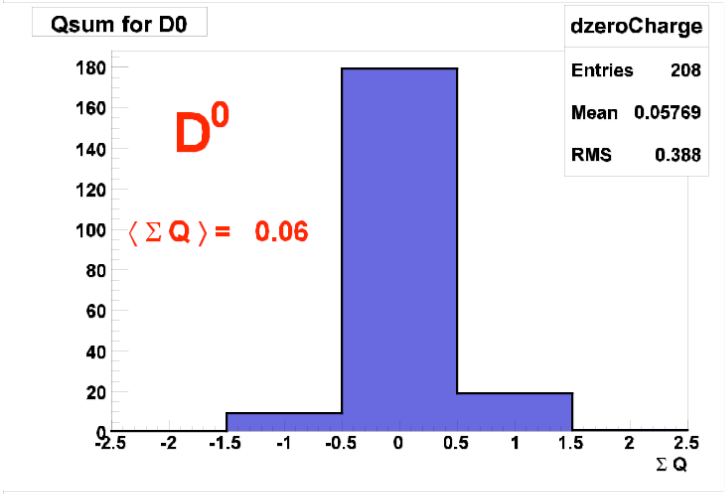
- results above were obtained for 100 GeV b-jets;
  corresponding studies at 25 GeV and 250 GeV jet energy show equally good agreement

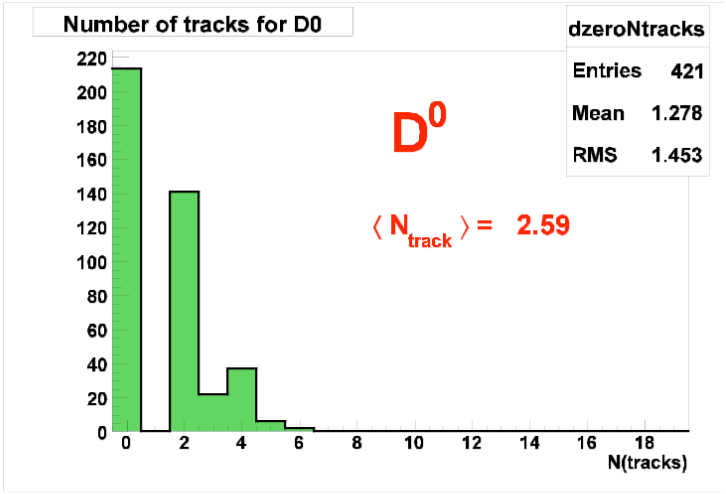
Vertex charge study for c-jets

- vertex charge (ZVTOP & L/D) and #(c decay tracks) for various decay channels (E_{CM} = 500 GeV)









Victoria Martin

- first results look very promising; next look at corresponding leakage rates and optimise L/D

Suggestion for a new Vertex class in LCIO

+~Vertex()

*+getMomentum() : const float**

+getMass() : float

+getCharge() : float

+getPosition() : const FloatVec&

+getCovMatrix() : const FloatVec&

+getChi2() : float

+getProbability() : float

+getDistanceToPreviousVertex() : float

+getErrorDistanceToPreviousVertex() : float

+getParameters() : const LCParameters&

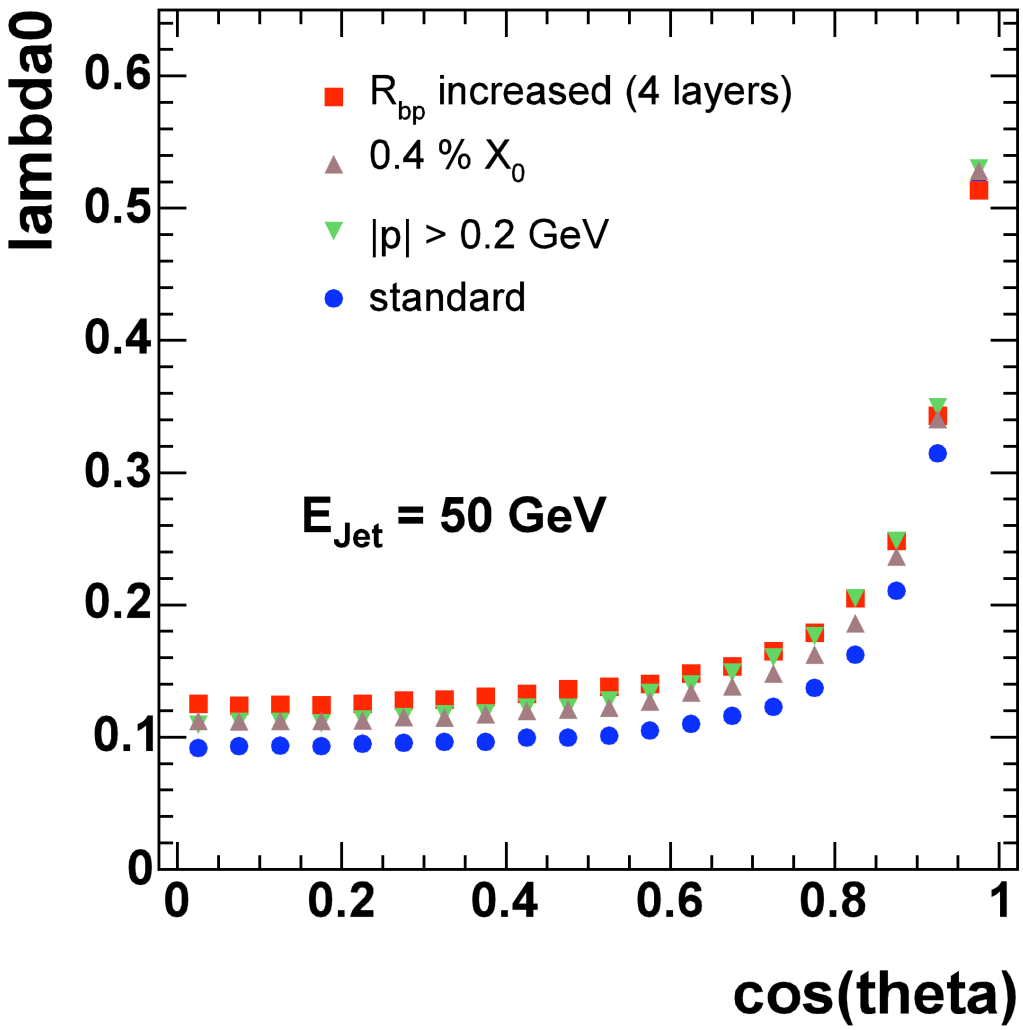
+getTracks() : const TrackVec&

+addTrack() : void

+getPreviousVertex() : Vertex&

- **proposed dedicated Vertex class to be included in LCIO (see left)**
- **negotiations with LCIO developers group are ongoing: may not be implemented in this form, but methods specific to Vertex class will be made available**

Sensitivity to other parameters



- since vertex charge performance is sensitive to multiple scattering need to keep layer thickness small (target 0.1 % X_0)
- also strong dependence on momentum cut (track selection) – this depends critically on tracking performance:
  - track finding capability
  - background rates
  - linking across subdetector boundaries

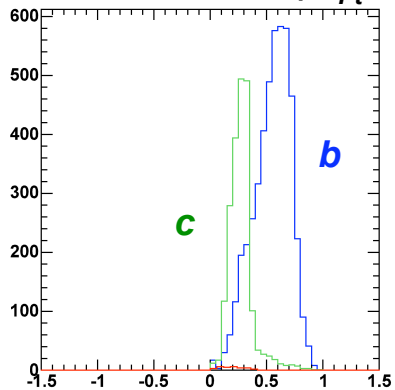
- should push all these parameters to their limits, as all these effects will eventually add up in the real detector

Flavour tag

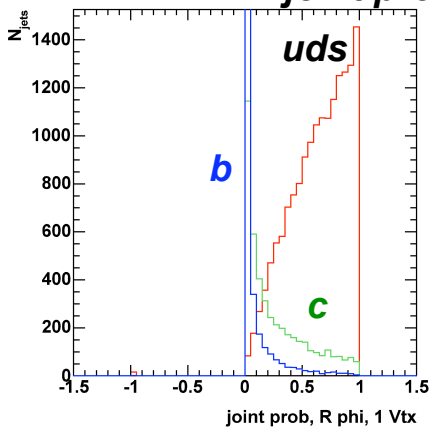
- Vertex package will provide flavour tag procedure developed by R. Hawkings et al (LC-PHSM-2000-021) and recently used by K. Desch / Th. Kuhl as default

- NN-input variables used:
  - if secondary vertex found: M_{Pt}, momentum of secondary vertex, and its decay length and decay length significance
  - if only primary vertex found: momentum and impact parameter significance in R-ϕ and z for the two most-significant tracks in the jet
  - in both cases: joint probability in R-ϕ and z (estimator of probability for all tracks to originate from primary vertex)

tanh (M_{Pt} / 5 GeV)



joint probability



- will be flexible enough to permit user further tuning of the input variables for the neural net, and of the NN-architecture (number and type of nodes) and training algorithm