Kinematic fitting in the presence of ISR

M. Beckmann, B. List, J. List

DESY

LCWS 2008 – Simulation and Reconstruction

Introduction
MarlinKinfit
Modelling ISR
Application to SM 4-jet events
Summary & Outlook

Motivation

- kinematic fitting uses known constraints of the events (e.g. $\sum p_x = 0$) to improve measured quantities and/or determine unmeasured quantities
- big improvement of jet energy resolution at LEP
- problem at ILC: ISR and Beamstrahlung [1]
- with naive approach: loose lots of events, bias di-jet masses for remaining ones

[1] „ISR" refers to both effects throughout the talk

Kinematic Fitting

- minimize $\chi^2 = (\vec{a} - \vec{\eta})^T Cov^{-1} (\vec{a} - \vec{\eta})$,
  $\vec{a}$: fitted parameters, $\vec{\eta}$: measured parameters
- apply "hard" constraints: $G_k(\vec{a}, \vec{\xi}) = 0$,
  $\vec{\xi}$: unmeasured parameters
- method: Lagrange multipliers: $\chi^2_T = \chi^2 + \sum \lambda_k G_k$,
  seek stationary point:
  $\frac{\partial \chi^2_T}{\partial a_i, \xi_j} = 0$ → best values of parameters
  $\frac{\partial \chi^2_T}{\partial \lambda_k} = G_k(\vec{a}, \vec{\xi}) = 0$ → constraints are fulfilled
- needs multidimensional equation solver → iterative method

MarlinKinfit: Software Structure

3 basic concepts:

- FitObject: e.g. JetFitObject
  - has several parameters, measured or unmeasured
  - can calculate χ^2 contribution and derivatives
  - can calculate 4-vector
- Constraint: e.g. MassConstraint
  - is formulated in terms of 4-vectors (from FitObjects)
  - can calculate constraint function G_k and derivatives
- Fitter: e.g. OPALFitter
  - manages the fitting problem, assigns numbers to parameters and constraints
  - sets up and solves system of equations

Code (with example) available as Marlin processor

Example Code

```
//                    E   theta phi    dE dtheta dphi mass
JetFitObjet jet1 (44., 1.2, 0.087, 5.0, 0.2, 0.1, 0.);
JetFitObjet jet2 (46., 1.8, 3.120, 5.0, 0.2, 0.1, 0.);

// Constraint 0*sum(E) + 1*sum(px) + 0*sum(py) + 0*sum(pz) = 0
Momentumconstraint pxconstraint (0, 1, 0, 0, 0);
pxconstraint.addToFOList (jet1);
pxconstraint.addToFOList (jet2);

OPALFitter fitter;

fitter.addFitObject (jet1);
fitter.addFitObject (jet2);

fitter.addConstraint (pxconstraint);

double prob = fitter.fit();
```

Soft Constraints

- “soft” constraints: Add additional terms to total χ^2
- example: $\Delta m(\vec{a}, \vec{\xi})$ is mass difference of two dijet-pairs
  $\rightarrow$ add $(\Delta m/\sigma)^2$ to total χ^2
- allows to take into account constraints that are not exactly fulfilled
- useful to avoid very small fit probabilities
- if unmeasured quantities are only constrained via “soft” constraints, some methods to solve equation system fail (e.g. OPALFitter)

Available Fit Engines

- OPALFitter
  - implementation of kinematic fit used at OPAL, (see L. Lyons: *Statistics for nuclear and particle physicists,* Cambridge Univ. Pr. 1986)
  - well tested, robust → our reference implementation
  - cannot treat case of soft constraints with unmeasured parameters
- NewtonFitter
  - new development, uses multidimensional Newton-Raphson method
  - can treat soft constraints with unmeasured parameters
  - needs (and uses!) 2nd derivatives of constraint functions → slightly better convergence
  - the plots in the rest of the talk are made with the NewtonFitter

SM 4-jet events

SLAC Whizard SM sample

- $e^+e^- \to u\bar{d}d\bar{u}$, 500 fb^{-1}
  (mostly W^+W^-, no ν from semileptonic c/b decays)
- full simulation of LDCprime
- full reconstruction, Particle Flow: Pandora
- force event into four jets (Durham4Jets on DST)
- require for each jet:

	events	
N_{tot}	52490	
$\lvert\cos\theta_{\text{jet}}\rvert < 0.989$	39940	76.1 %
$N_{\text{track}}/\text{jet} \geq 1$	38925	74.2 %
$E_{\text{jet}} > 4.5$ GeV	38912	74.1 %

ISR classification

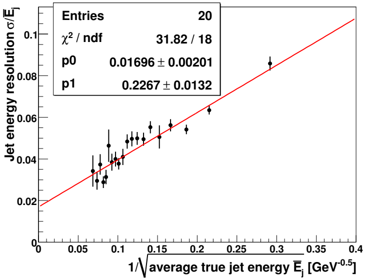
select (MC info) events with either

- no significant ISR:
  $\sum E_\gamma < 5$ GeV $\rightarrow$ 24731 ev. (64 %)
- with significant ISR:
  $\geq 1\ \gamma$ with $E_\gamma > 25$ GeV, $\cos\theta_\gamma > 0.999$ $\rightarrow$ 6699 ev. (17 %)
- later: $\rightarrow$ 7482 ev. (19 %)
  - photons within detector acceptance
  - „soft“ ISR

Jet Parametrisation

roughly determined from „no ISR“ sample:

- Jet energy scale: $E = 1.01 \cdot E_{\mathrm{rec}}$
- $\delta E = 23\% \cdot \sqrt{E} + 1.7\% E$ (from $u\bar{d}d\bar{u}$ sample, not single hadrons)
- $\delta\theta = 0.01\,\mathrm{rad}$
- $\delta\phi = 0.01\,\mathrm{rad}$



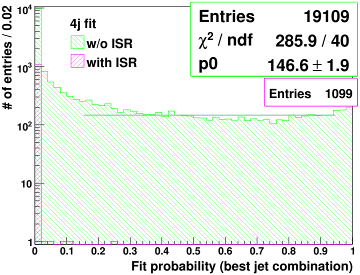
4 Jet Hypothesis

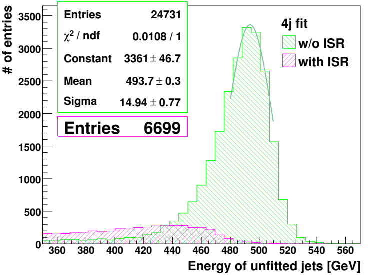
Constraints

- $\sum E^{\rm jet} = E_{\rm CM}$
- $\sum p_x^{\rm jet} = 0$
- $\sum p_y^{\rm jet} = 0$
- $\sum p_z^{\rm jet} = 0$
- equal di-jet masses

...as expected:

- works well on events w/o ISR
- hopeless for ISR sample....





Modelling ISR

Soft Constraints

- don't force $E_{\mathrm{CM}} = 500$ GeV and $\sum p_z = 0$
- use „soft" constraints instead,
  i.e. no Lagrange multiplier but just additional χ^2 contribution depending on deviation from naive beam constraint
- problem: need to keep E_{CM} and $\sum p_z$ consistent!
- could constrain $E_{\mathrm{CM}} \pm \sum p_z$ ....
- in combination with unmeasured parameters (ν, LSP):
  minimisation get's more involved, need special fit engine!
- currently not pursued

Modelling ISR

PhotonFitObject, „unmeasured“

- treat photon parameters as unmeasured
- this „costs“ constraints
- usually not applicable for final states with true E_{miss} (ν, LSP)

PhotonFitObject, measured

- treat photon parameters as measured
- choose error parametrisation such that fitted E_γ spectrum reproduces ISR spectrum
- advantage: no loss of constraints!

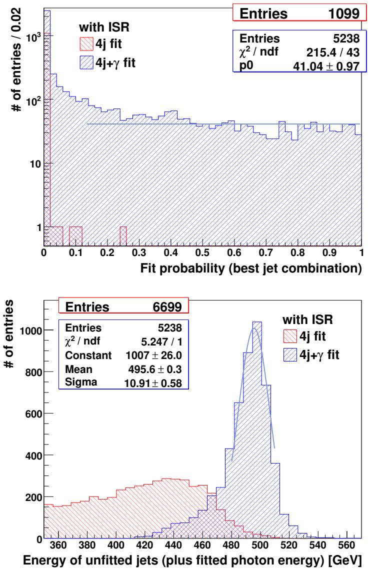
Application to SM 4-jet events

Photon Parametrisation

- one parameter: p_z
- others fixed via $E = |p_z|,\ p_x = p_y = 0$
- treat p_z as *measured*!
- with currently Gaussian error: $p_z = 0 \pm 100$ GeV

„with ISR " sample

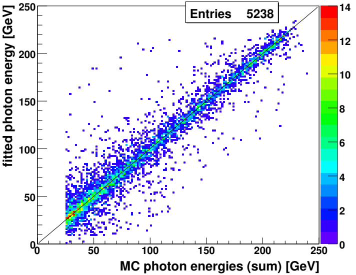
- vast improvement!
- converged fits 78%
- fit prob flat, $E_{\text{tot}} \simeq 500$ GeV

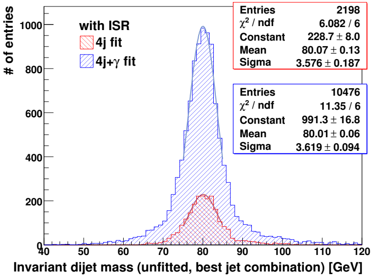


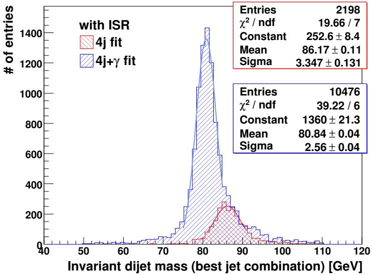
4 Jet + Photon Hypothesis

„with ISR “ sample

- nice correlation between E_γ^{rec} and E_γ^{MC}
- no shift in mass peak
- **$\sigma_{M_{jj}}$: 3.6 GeV → 2.6 GeV**



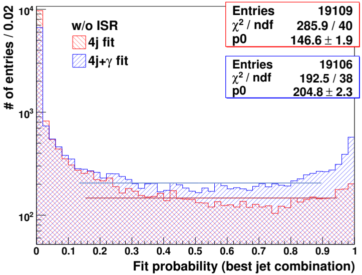


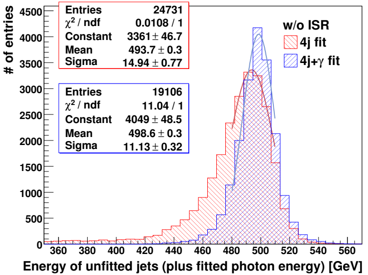


4 Jet + Photon Hypothesis

„no ISR " sample

- higher fit probabilities
- total reconstructed energy shifts by $\simeq$ 5 GeV
  $\rightarrow$ not too bad, since up to 5 GeV ISR allowed also in this sample

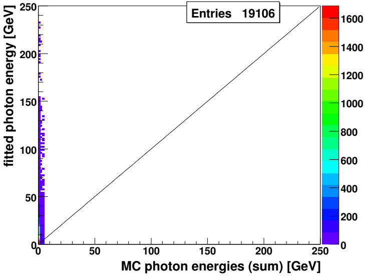


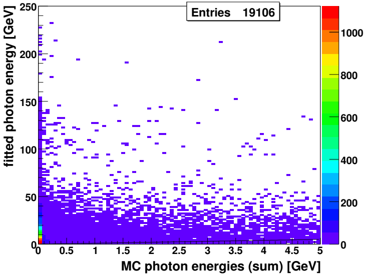


4 Jet + Photon Hypothesis

„no ISR " sample

- majority of events gets only little fake photon energy
- some events with huge fitted photon energies
  $\rightarrow$ investigate...
- improvement expected with
  - proper probability density describing ISR spectrum
  - optimized jet error parametrisation

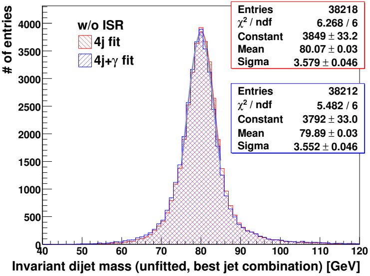


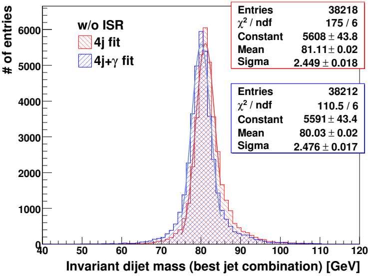


4 Jet + Photon Hypothesis

„no ISR “ sample

- convergence: 77% for both fits
- mostly same jet pairing „best“
- 4 jet fit: $< M_{jj} >= 80.1$ GeV $\rightarrow$ 81.1 GeV
- nearly no bias by 4j+γ fit: $< M_{jj} >= 80.0$ GeV
- $\sigma_{\mathbf{M_{jj}}} =$ **2.5 GeV for both fits**





Summary

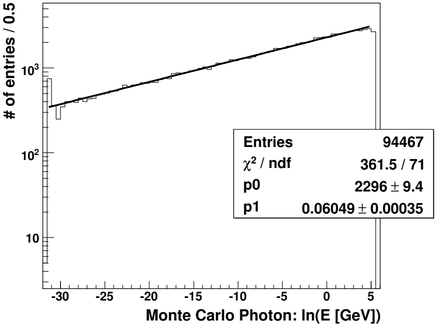
- di-jet mass resolution can be significantly improved by kinematic fits even in the presence of ISR
- already with a very basic photon parametrisation significant amounts ISR energy can be well reconstructed
- many details not yet optimized
  $\Rightarrow$ possibilities for improvement!
- SM $u\bar{d}d\bar{u}$ events with and without ISR:

  $$\sigma_{\mathbf{M_{jj}}} = \mathbf{3.6\ GeV} \rightarrow \mathbf{2.6\ GeV}$$

- fitting with ISR hypothesis:

  same resolution gain as w/o ISR,
  but **without any bias** on reconstructed dijet mass!
- PhotonFitObject will be in next MarlinReco release

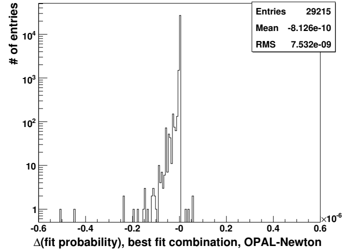
Outlook

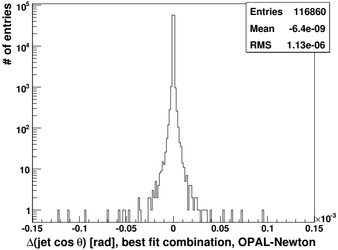


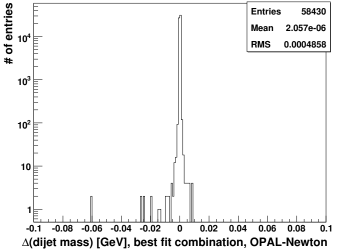
- serious photon parametrisation
- include events with little ISR
- look at events with photon in detector (5 jet collection?)
- try other quark flavours
- optimize jet energy scale, jet error parametrisation, convergence criteria, ...
- maybe look at soft constraint approach for comparison
- apply to Chargino / Neutralino separation
- write LCNote and Dipoma Thesis (Moritz :-)

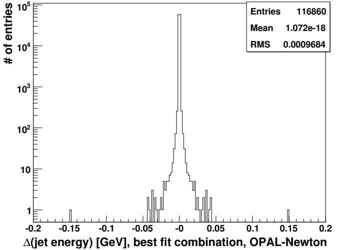
BACKUP

OpalFitter vs NewtonFitter









OpalFitter vs NewtonFitter