The BACMATCH algorithm

Summerstudent Program 2007, DESY

Iurii Sorokin[a]

Superviser:

Achim Geiser[b]

[a] National Taras Shevchenko University of Kyiv, Ukraine

[b] Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany

14th September 2007

Abstract

The BACMATCH algorithm is intended to match tracks in BAC with tracks in the inner parts of the ZEUS Detector.

Contents

1 **Introduction** **2**

2 **Essential parts of the Detector** **3**

2.1 Tracking system (TS) 3

2.2 Calorimeter (CAL) 3

2.3 Backing Calorimeter (BAC) 3

3 **The BACMATCH Algorithm** **4**

3.1 Algorithm overview 4

3.2 Source files 4

3.3 BACMATCH Input and Output. Interplay with ZEUS analysis framework 5

3.4 Track representations 6

3.5 Technical realization 6

4 **Results and conclusion** **8**

5 **Acknowledgements** **8**

1 Introduction

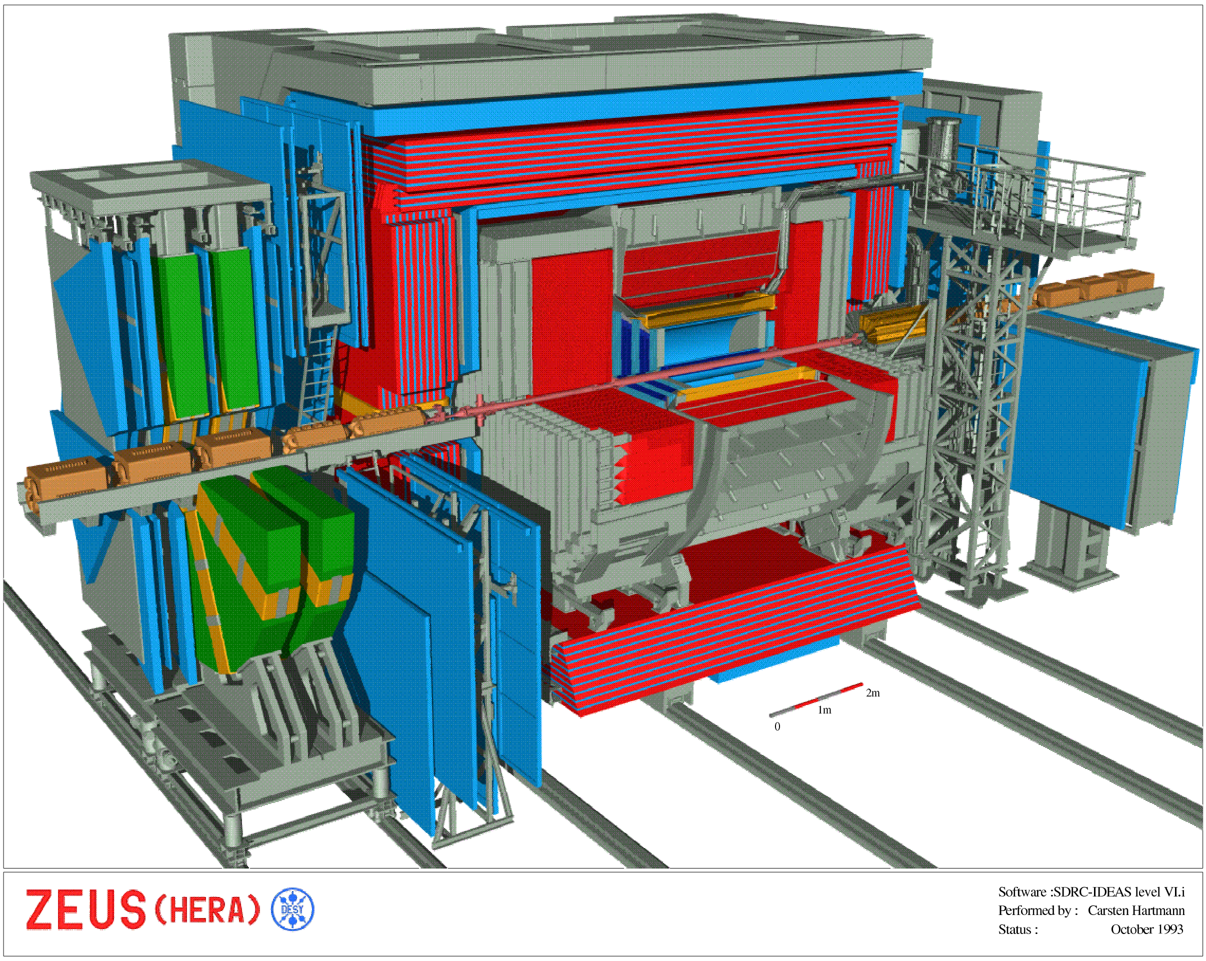

Figure 1: *The ZEUS detector.*

The ZEUS Detector [1] is a general purpose magnetic detector designed to study the different processes of lepton-proton scattering at the HERA collider.

The ZEUS Detector Tracking System(TS) provides a precise measurement of track momenta as well as track positions. The Backing Calorimeter (BAC) is also suited for muon tracking. The BACMATCH algorithm, described in this work, is intended to match tracks in the TS and BAC. Improving the BAC information used in standard analyse will significantly decrease the muon backgruond.

At present, BACMATCH is still in development. It is not included in released analysis software and could not be used efficienly yet.

2 Essential parts of the Detector

The ZEUS Detector is very complex, thus only parts essential for BACMATCH are described.

2.1 Tracking system (TS)

The Central Tracking Detector (CTD) [2] provides measurements of track momentum and position. The Silicon Microvertex Detector [3] improves the track position measurment with a precision better than 50 μm close to the beamspot. Thus the MVD allows to find vertices very efficiently.

2.2 Calorimeter (CAL)

The CAL [4] is a high resolution calorimeter consisting of plates of depleted uranium interleaved with plastic scintillator as an active material. The calorimeter provides precise energy measurements for hadrons and jets. Muons, passing through the Calorimeter, lose significant amount of energy.

2.3 Backing Calorimeter (BAC)

The Backing Calorimeter [5] consists of proportional chambers which are located between the iron sheets of Yoke. There are 9 layers of BAC chambers alternating with 10 layers of steel. It covers the whole ZEUS detector, with the exception of the Beam Pipe holes.

The proportional chambers are assembled from 8 or 7-tube wide aluminum modules. Since BAC has no high resolution time readout, the precision of the hit position measurment is defined by chamber size. The Anodes (wires) have lengths up to a few meters. The Cathodes (pads) are planes of about 25x25 cm. Thus, using few hits, BAC allows to measure a muon track position and direction with high precision along two axes of reference, and roughly along the third one.

The pires have digital readout only. The pads has analog readout that allows to measure the shower energy in case of leakage from the CAL. That is why, BAC is called a calorimeter.

3 The BACMATCH Algorithm

3.1 Algorithm overview

The present alorithm repeats the idea of the BREMAT [6] algorithm (by G. Abbiendi), which matches inner tracks with hits in the Muon Chambers. The idea is to extrapolate tracks from the inner detector to the BAC. BACMATCH uses the GEANE v3.11 [7] package for track extrapolation. Making the track propagation, we take into account energy losses and calculate the full error matrix. If the extrapolated track crosses BAC close enough to any BAC track, we try to give a quality of matching, like χ^2. Then all information about every matched pair is stored. The alghorithm is not meant to finally decide which inner track corresponds to a BAC track. It is meant to give a quantative estimation of the accordance of each potentially matching track.

At present, the BACMATCH algorithm does track extrapolation, but doesn't yet give a quality of matching.

3.2 Source files

At present, BACMATCH is in development. The source files are located in the same directories as those of the BREMAT alorithm. Soon, they will be placed to a separate directory.

BACMATCH algorithm is intended to be a part of the PHANTOM[8] library. So it is written in Fortran. Source files are:

- *PHANTOMDIR/src/Muons/bremat/bacmatch.fpp* — the main BACMATCH routine
- *PHANTOMDIR/src/Muons/bremat/eustep_bacmatch.fpp* — track propagation control routine. Called through the chain of routines:
  `BACMATCH` → `ERTRAK` → `ERTRGO` → `EUSTEP` → `ZGEAST` → `EUSTEP_BACMATCH`
- *PHANTOMDIR/src/Muons/bremat/brebac.inc* — defines BACMATCH ORANGE block structure. Will be renamed to *bacmatch.inc* soon.
- *ORANGEDIR/src/ntuple/nt_define/define_brebac.fpp* — creates BACMATCH block branches in ROOT tree (or PAW ntuple). Generated automaticaly by *ntgen* [11] according to *brebac.inc*

To provide correct functioning of the algorithm and it's interplay with the other analysis environment, the following modifications to existing files were done:

- *EAZE_JOBDIR/cmd/makefile.preamble* — paths to PHANTOM and ORANGE were specified

- *PHANTOMDIR/src/Muons/bremat/breana.fpp* — added `call BACMATCH` statement at the end of the file, just before `return` statement. In the future, `BACMATCH` will be called directly from ORANGE

- *EAZE_JOBDIR/cmd/control.cards* — GEANE cards added, BACMATCH block turned ON, minor changes. By now, BACMATCH block named BREBAC, but will be renamed to BACMATCH soon.

- *ORANGEDIR/ntgen.input* — definition of BACMATCH(BREBAC) block added.

- *ORANGEDIR/src/ntuple/nt_define/define_brebac.fpp* — in `#include` statement, absolute path to *brebac.inc* was specified.

- *ORANGEDIR/src/ntuple/user_nt_define.fpp* — made to contain only one executable line: `call define_brebac`

- *GLTRKDIR/src/geanes_tracking/zgeast.fpp* — In case of `BREMAT_flag` equals to 2 `eustep_bacmatch` called.

3.3 BACMATCH Input and Output. Interplay with ZEUS analysis framework

To make track matching, the next information is necessary:

- ZEUS Detector geometry, material arrangement, magnetic field map are needed by GEANE for track extrapolation. Since BACMATCH is a part of PHANTOM library, at the moment of `BACMATCH` routine call, GEANE is already initialized, material arrangement is loaded. Magnetic field could be obtained through special routine `GUFLD`

- Tracks, reconstructed with TS. They are available in ADAMO [9] tables.

- BAC tracks and geometry. Reconstructed tracks are read from `MUBAC_VAR` common block. The spatial orientation of the BAC sensitive planes is taken from ADAMO.

BACMATCH uses an ORANGE [10] block of the same name for data output.

3.4 Track representations

The particle trajectory is characterized by 5 independent variables as a function of one parameter (e.g. the path length). Among the 5 variables one is related to the curvature (to the absolute value of the momentum, p), two are related to the direction of the particle and the other two are related to the spatial location. BACMATCH uses three representations of these 5 parameters:

VC representation:

$$\phi, \quad \frac{Q}{p_T}, \quad QD, \quad z, \quad \cot\theta$$

ϕ — azimuthal angle (angle tangent to the helix in ZEUS xy plane)
θ — angle of dip with regards to ZEUS xy plane ($\theta \equiv \frac{\pi}{2} - \lambda$)
Q — charge
p_T — transverse momentum
D — distance to the center of ZEUS reference frame.
z — z coordinate in ZEUS reference
Using this representation, `VCATCAL` table error matrix is given.

SC representation:

$$\frac{1}{p}, \quad \lambda, \quad \phi, \quad y_\perp, \quad z_\perp$$

where p is the track momentrum,
λ and ϕ are the dip and azimuthal angles related to the momentum components in the following way:

$$\begin{aligned} p_x &= p\cos\lambda\cos\phi \\ p_y &= p\cos\lambda\sin\phi \\ p_z &= p\sin\lambda. \end{aligned}$$

$y_\perp$ and $z_\perp$ are the coordinates of the trajectory in a local orthonormal reference frame with the $x_\perp$ axis along the particle direction, the $y_\perp$ being parallel to the xy plane. GEANE output error matrix is given in this representation.

SD representation:

$$\frac{1}{p}, \quad v', \quad w', \quad v, \quad w$$

with $v' = dv/du$ and $w' = dw/du$ in an orthonormal coordinate system with axes u, v and w. GEANE input error matrix has to be given in this representation.

3.5 Technical realization

The source code is described superficially, see *bacmatch.fpp* for details.

The `BACMATCH` routine is called for each event. First it checks if the event is not a BOR (Beginnig Of Run). If it is a first execution, `BACMATCH` reads rotation matrices of BAC parts from the `XRot` ADAMO table. They are stored in the array `real RotBAC(3,3,17)`. The `save RotBAC` statement compels Fortran to remeber the matrices, such that they can be read more than once.

Then BACMATCH loops over TS tracks, which are taken from the `VCATCAL` ADAMO table. `VCATCAL` describes the track state, at the moment it reaches CAL. The track is being extrapolated, if it satisfies a few criteria. The most important, is that the track has to have enough momentum, to reach BAC. Otherwise, extrapolation is senseless.

Once a TS track passed the preselection, BACMATCH loops over BAC tracks to find matches for it. BAC tracks are read from the `MUBAC_VAR` common block. The access to the common block is provided by including *mubac.inc*. Some cuts are also applied on BAC tracks.

To obtain the spatial orientation of the BAC element which detected a track, BACMATCH uses the so-calledGG index — a unique identification number of the BAC sensitive element. The `XXLay` ADAMO table provides the correspondence between the GG index of the BAC element and it's rotation matrix. Similar ADAMO table for Pads exists, but is not being filled yet. Thus, at present, it is impossible to obtain the orientation of a BAC element without wire readout (possible by implication). BAC tracks without wire readout are being skipped so far.

To extrapolate a track, GEANE requires it's initial state. The track initial coordinates and momentum have to be provided as 3-vectors. The error matrix is expected to be given in the SD representation. Also a local reference frame orientation for the SD representation has to be provided.

To convert the error matrix from VC to SD, BACMATCH converts it first to the SC representation with `BREVCSC`. And then, from SC to SD using the `TRSCSD` routine.

Togeter with the track initial state, the user has to tell GEANE, when the track extrapolation has to be stopped. GEANE supports a few criteria of track extrapolation breaking. BCMATCH runs GEANE in the mode of extrapolation, until the predefined plane is reached. The plane is defined with the `EUFILP` routine. Track extrapolation is initated by calling `ERTRAK`.

By now, BACMATCH doesn't yet calculate the χ^2 of the matching. It just calculates the distance between the measured track and the predicted one (for development purposes).

If `BREDEBUG` > 0, the routine prints lots of debugging information (for experts only).

4 Results and conclusion

In its present stage of development, BACMATCH already does track extrapolation, writes some debuging information to it's ORANGE block and eaze_job log file. Tests on a MC dimuon sample showed, that the extrapolation is correct. It's going to be finished in the nearest future and being implemented as a separate PHANTOM module.

5 Acknowledgements

I would like to thank Achim Geiser for his guidance, ideas and help, Grzegorz Grzelak for help with BAC, Monica Turcato for help with BREMAT, Giovanni Abbiendi for BREMAT algorithm existence, Ingrid Gregor for organizing different arrangements for ZEUS summerstudents, Vladimir Aushev for his encouragement

Also, I would like to thank Joachim Meyer, Elisabetta Gallo, Tobias Haas and all the DESY staff for the great summerstudent program.

References

[1] The ZEUS Detector: Status Report 1993 *The ZEUS Collaboration*

[2] Nucl. Instr. and Meth. A338, 254 (1994) *E. Foster et al.,*

[3] A. Garfagnini, Nucl. Inst. and Meth. **A435**, 34 (1999)
C. Coldewey, Nucl. Inst. and Meth. **A435**, 149 (2000)
E.N. Koffeman, Nucl. Inst. and Meth. **A435**, 89 (2000)
U. Koetz, Nucl. Inst. and Meth. **A461**, 210 (2001)
E.N. Koffeman, Nucl. Inst. and Meth. **A473**, 26 (2001)
V. Chiochia, Nucl. Inst. and Meth. **A501**, 60 (2003)
D. Dannheim *et al.*, Nucl. Inst. and Meth. **A505**, 663 (2003)

[4] A. Andresen *et al.* (ZEUS Calorimeter Group), Nucl. Inst. and Meth. **A309**, 101 (1991)
M. Derrick *et al.* Nucl. Inst. and Meth. **A309**, 77 (1991)
A. Caldwell *et al.* Nucl. Inst. and Meth. **A321**, 256 (1992)
A. Bernstein *et al.* Nucl. Inst. and Meth. **A336**, 23 (1993)

[5] T. Jezynski *et al.* Proc. SPIE 5484, 180 (2004) K. Pozniak *et al.* Proc. SPIE 5484, 186 (2004)

[6] G. Abbiendi ZEUS-Note 99-063, 1999

[7] V. Innocente, M. Maire and E. Nagy, "GEANE: average tracking and error propagation package", proc. of MC91: Detector and event simulation in high energy physics (Amsterdam, 1991).

[8] http://www-zeus/ZEUS_ONLY/analysis/Phantom/index.html

[9] M.G. Green ZEUS-Note 89-027, 1989
A. J. Sephton ZEUS-Note 89-108, 1989
http://www-zeus-data.desy.de/zeusddl.pdf
http://adamo.web.cern.ch/Adamo/ADAMO_ENTRY.html

[10] http://www-zeus/ZEUS_ONLY/analysis/orange/index.html

[11] http://www-zeus.desy.de/ZEUS_ONLY/analysis/orange/
using_and_extending_the_orange_ntuple.html