

KOPIO TN089v3

# FastMC User Manual

David E. Jaffe, BNL

October 20, 2004

## **Abstract**

The FastMC functionality and usage is described.

# Contents

<b>1</b>	<b>Overview</b>	<b>6</b>
1.1	Brief description . . . . .	6
1.2	Units and coordinate system . . . . .	6
1.3	Installation . . . . .	7
1.4	Directory structure . . . . .	7
1.5	Making the executables . . . . .	7
1.6	Running jobs . . . . .	7
1.6.1	<code>simple.run.fastmc.script</code> . . . . .	8
1.6.2	Standard script for production jobs <code>run.fastmc.script</code> . . . . .	8
1.6.3	<code>submit.many.fastmc.script</code> . . . . .	8
1.6.4	<code>submit.ALL.fastmc.script</code> . . . . .	8
1.6.5	<code>submit.standard.bunch.of.jobs.script</code> . . . . .	9
1.6.6	Analysis of ntuples with <code>analysis.script</code> . . . . .	9
1.6.7	<code>Trim.Down.ANAL</code> . . . . .	9
1.6.8	<code>Trim.to.File</code> . . . . .	10
1.7	Control files . . . . .	10
1.8	Geometry files . . . . .	10
1.9	Production of ntuples . . . . .	14
<b>2</b>	<b>Track propagation and decay weights</b>	<b>14</b>
<b>3</b>	<b>The stack and source/<code>trackinfo.inc</code></b>	<b>14</b>
<b>4</b>	<b>Fake and merged photons</b>	<b>15</b>
4.1	Fake photons from stopped muons . . . . .	16
4.2	Fake photons from neutron-induced showers . . . . .	16
4.3	Merging of photons . . . . .	17
<b>5</b>	<b>Neutron propagation and interaction</b>	<b>18</b>
<b>6</b>	<b><math>K_L^0</math> spectrum and production rate</b>	<b>18</b>
<b>7</b>	<b><math>K_L^0</math> Production point and beam aspect ratio</b>	<b>18</b>
<b>8</b>	<b>Smearing of hits</b>	<b>19</b>
8.1	Contrasting Bryman and Zeller PR models . . . . .	19
<b>9</b>	<b>Reconstruction and fitting</b>	<b>20</b>
<b>10</b>	<b>Changing the radiation lengths of the PR</b>	<b>21</b>
<b>11</b>	<b>How yields are calculated in <code>analysis.script</code></b>	<b>22</b>
<b>12</b>	<b>More details of candidate weight calculation</b>	<b>23</b>

<b>13 Possible improvements/modifications</b>	<b>24</b>
<b>14 Acknowledgements</b>	<b>25</b>

# List of Tables

1	Description of arguments to <code>run.fastmc.script</code> . The detector file is assumed to be in the <code>geometry/</code> directory. The decay file is assumed to be in the <code>scripts/</code> directory. See Section 1.6 and 2, 3 and Tables 4 for further description of arguments. . . . .	10
2	Part 1 of key words, format and description used in control file. Format of “I” means integer, “R” means real and “C” means character string. All variables <b>must</b> be given even if the values are irrelevant. . . . .	11
3	Part 2 of key words, format and description used in control file. Format of “I” means integer, “R” means real and “C” means character string. All variables <b>must</b> be given even if the values are irrelevant. “SKIM” requirements are ignored if skimming is not activated. The key words “MAXIMUM COS(K,e) FOR KE4”, “FORCE YBEAM HEIGHT IN FIT” and others are written in a small font so that I can cram them into the columns of this table. . . . .	12
4	The upper part of the table lists the number, four-character names of the allowed $K_L^0$ decay modes in the FastMC. A reasonable approximation to the correct matrix element is used for all $K_L^0$ decays except for $K_L^0 \rightarrow K^\pm e^\mp \nu$ which uses phase space. The lower half lists the other particle decays, mode numbers and branching fractions. The rightmost column indicates whether the matrix element (ME) or phase space (PS) is used for the decay dynamics. . . . .	13
5	Particle number and type for particles in the FastMC. A “merged photon” can be created from a pair of real or fake photons. . . . .	15

# List of Figures

1	A sample file with an ntuple list used as input to <code>analysis.script</code> . See Section 1.6 for file naming conventions. . . . .	9
2	The $ X $ and $ Y $ fiducial limits on candidate photons. A value less than zero means no limit is applied. . . . .	24
3	The photon veto inefficiencies as a function of photon energy in MeV/c used in analysis. The standard photon veto inefficiency as well as approximations to the veto inefficiency of E787 and KAMI are shown. . .	26
4	The explicit form of the standard PV function. . . . .	27
5	Part 1 of the explicit form of the standard charged veto functions. . . .	28
6	Part 2 of the explicit form of the standard charged veto functions. . . .	29
7	The charged particle veto inefficiencies as a function of momentum in MeV/c used in analysis. . . . .	30
8	The parametrized neutron energy spectrum. . . . .	31
9	Part 1 of the file <code>scripts/control.dat</code> . . . . .	32
10	Part 2 of the file <code>scripts/control.dat</code> . . . . .	33
11	Part 3 of the file <code>scripts/control.dat</code> . . . . .	34
12	The file <code>condir/control.kp2.21064.21</code> created by <code>scripts/run.fastmc.script</code> for $K_L^0 \rightarrow \pi^0\pi^0$ production. . . . .	35
13	Part 1 of the file <code>geometry/detector.21.dat</code> . . . . .	36
14	Part 2 of the file <code>geometry/detector.21.dat</code> . . . . .	37
15	Part 3 of the file <code>geometry/detector.21.dat</code> . . . . .	38
16	Per-particle info stored in <code>trackinfo.inc</code> . . . . .	39
17	The first part of the per-candidate information. . . . .	40
18	The second part of the per-candidate information. . . . .	41
19	Third part of the per-candidate information. . . . .	42

# 1 Overview

Abbreviation	Description
PR	preradiator
CAL	calorimeter
US	upstream
DS	downstream
BV	barrel veto
OV	PR outer veto

## 1.1 Brief description

- Simple geometry: all volumes are rectangular parallelepipeds
- No magnetic fields
- Several models of PR resolution available (Section 8.1)
  - “Konaka” - analytic form of position, angular resolution (double gaussians)
  - “Zeller” - simulation of pair creation, brems. in WC geometry similar to proposed geometry
  - “Bryman” - improved Konaka model, has energy and angular dependence of resolution
- Two stage process
  1. Generation (Single- or multi- $K_L^0$ , single  $nN \rightarrow \pi^0 X$ ) (program file `source/fastmc.f`)
    - Smearing of 'hits' (file `source/smear.f`)
    - Reconstruction of  $K_L^0$  candidates (file `source/recon.f`)
    - Skim (optional) (file `source/skimevent.f`)
    - Output ntuple
  2. Analysis from ntuple (program file `source/anal.f`)
    - Hermiticity assumed, except for upstream beam hole.
    - $\bar{\epsilon}_{PV}(E_\gamma)$  for photon veto assumed.
    - $\bar{\epsilon}_{CV}(E, \text{species})$  for charged veto assumed.
    - Veto timing ignored thus far.

## 1.2 Units and coordinate system

The units of the FastMC are centimeter, **MeV** and **nanosecond**. Note that GEANT uses centimeter, **GeV** and **second**. A right-handed coordinate system is used with  $z = K_L^0$  beam direction and  $y = \text{up}$ . The origin is the center of the production target.

## 1.3 Installation

Documentation and instructions for downloading the FastMC via CVS can be obtained from <http://www.phy.bnl.gov/~djaffe/KOPIO/fastmc/>.

## 1.4 Directory structure

The following directory structure is also described in `README` in the main directory.

Directory	Description
analysis	for paw-based analysis using Akira-style ntuples
DataBase	material info and skim stats
geometry	detector definition
scripts	run jobs with these
source	code and executables
Templates	templates used by scripts to make control files

Here are additional directories used by scripts, since their contents can be large, the user should create them. Use `RUNME` in the main directory for default creation.

Directory	Description
condir	archived control files for jobs
log	for log files
ntuples	for ntuples and hist
locallog	for log files, I softlink it to log directory
localntuples	for ntuple and hist files, I softlink it to ntuples directory
exe	for executables

## 1.5 Making the executables

Before making executables you need to define the environmental `$CERNLIB` that gives the location of the CERN libraries via `setenv CERNLIB /cern/pro/lib`, for example.

1. in `source/`
2. `make` (creates `../exe/fastmc.exe` the executable for event generation)
3. `make -f Makefile.anal` (creates `../exe/anal.exe`, the executable for analysis)

## 1.6 Running jobs

Additional documentation is provided in the scripts themselves. All scripts are in the directory `scripts/`.

The scripts reference input files that define the geometry (i.e, `geometry/detector.21.dat`), control the program (i.e., `scripts/control.dat` or `Templates/control.template`), and accumulate the job statistics (`DataBase/skim.statistics.and.log`). Scripts produce output log files that are put into directory `locallog/` with a name containing the date and time of the job. Ntuples and histograms produced by scripts that generate

events are stored in directory `localntuples`. Histograms produced during analysis by `scripts/analysis.script` are stored in directory `ntuples`.

### 1.6.1 `simple.run.fastmc.script`

Handy for test or special production jobs.

### 1.6.2 Standard script for production jobs `run.fastmc.script`

- Arguments are described in Table 1.
- Input control file is created from arguments and template (see Section 1.7).
- Input geometry is given as an argument.
- Second random seed is created from arguments 5,6 and 7 (see below).
- Output log and ntuple file names are created from arguments and time of job start.
- The ntuple file name format is `ntuples/ModeName.Seed1.Seed2.rzn.entskim` where
  - `ModeName` is the four-character name (Table 4),
  - `Seed1` is the first random seed (and run number),
  - `Seed2` is the second random seed and is defined as  $100 \times Arg_7 + 10 \times Arg_6 + Arg_5$  with  $Arg_i$  as the  $i^{\text{th}}$  argument,
  - `rzn` is the default suffix, and
  - `entskim` is appended if skimming is enabled (Table 3).
- An example ntuple file name is `../ntuples/kcp3.61052.21.rzn.entskim` and the corresponding the log file is `../log/kcp3.61052.21.070504.0956.log` where `070504.0956` is the date and time in the format `ddMMyy.hhmm` with
  - `dd`=day,
  - `MM`=month,
  - `yy`=year,
  - `hh`=hour,
  - `mm`=minute.

### 1.6.3 `submit.many.fastmc.script`

For a bunch of production jobs, uses `run.fastmc.script`

### 1.6.4 `submit.ALL.fastmc.script`

For ALL production jobs, uses `run.fastmc.script`

```

../ntuples/kpnn.11054.21.rzn.entskim
../ntuples/kp2.21064.21.rzn.entskim
../ntuples/kp3.51054.21.rzn.entskim
../ntuples/ke3g.91053.121.rzn.entskim
../ntuples/kcp3.61052.21.rzn.entskim
../ntuples/ke4.121060.21.rzn.entskim
../ntuples/kpgg.111051.21.rzn.entskim

```

Figure 1: A sample file with an ntuple list used as input to `analysis.script`. See Section 1.6 for file naming conventions.

### 1.6.5 `submit.standard.bunch.of.jobs.script`

More up-to-date version of `submit.ALL.fastmc.script`, uses `run.fastmc.script`. Submits multiple jobs for  $K_L^0 \rightarrow \pi^0 \nu \bar{\nu}$  and “standard”  $K_L^0$  background modes. See script for details.

### 1.6.6 Analysis of ntuples with `analysis.script`

Jaffe-style ntuples output from `run.fastmc.script` can be analyzed by `analysis.script`. Typical usage is

$$\text{analysis.script ntuple.list (geometryfile)} \quad (1)$$

where `ntuple.list` is a list of ntuples produced by `run.fastmc.script` (See Figure 1 for an example) and the optional second argument gives the full path to the geometry file (i.e. `../geometry/detector.21.dat`). The analysis job assumes that the mode name is part of the ntuple file name as described in Section 1.6.

Analysis options are mainly controlled by `source/setupanal.f`. Recompilation is usually necessary for changing analysis.

- Jaffe-style ntuple variables are described in
  1. `trackinfo.inc` : MC truth and smeared hit info (see Figure 16)
  2. `reconinfo.inc` : reconstruction info (see Figures 17, 18 and 19)
  3. `trigger.inc` : very rudimentary hit counting
- Akira-style ntuple variables are listed in
  1. `kinema.cmn` : MC truth
  2. `kinfit.cmn` : reconstruction info

### 1.6.7 `Trim.Down.ANAL`

Produce compact output of analysis job from logfile.

Argument	Description (default value for optional arguments)
1	Job name, usually the four-character decay mode name. Table 4
2	Decay mode number. Table 4
3	Events to generate
4	Run number, also used as random seed 1
The following arguments are optional	
5	ANNIHILATION option (1)
6	CONVERSION POINT (3)
7	USE K WEIGHT (1)
8	detector file ( <code>detector.dat</code> )
9	decay file ( <code>default.decay.dat</code> )
10	MAXIMUM COS(K,e) FOR KE4 (1.1)
11	SKIM ACTIVATE (0)
12	FORCE YBEAM HEIGHT IN FIT (-10.)

Table 1: Description of arguments to `run.fastmc.script`. The detector file is assumed to be in the `geometry/` directory. The decay file is assumed to be in the `scripts/` directory. See Section 1.6 and 2, 3 and Tables 4 for further description of arguments.

### 1.6.8 Trim.to.File

Uses `Trim.Down.ANAL` to direct compact output to a file.

## 1.7 Control files

Control files are ascii files that allow the user to select the  $K_L^0$  decay mode, the Z range of  $K_L^0$  decays, the hit-smearing option, etc. All key words and options are described in Tables 2 and 3. The vast majority of these inputs for these key words are echoed in some form in the log file at the beginning of the job. Code to check for some obvious conflicts has been implemented and will stop the job with an error message.

The script `scripts/simple.run.fastmc.script` reads `scripts/control.dat` that is shown in Figure 9,10,11. In general this script and control file should be used only for tests.

The script `scripts/run.fastmc.script` uses `Templates/control.template` to build a control file based on the input arguments to `scripts/run.fastmc.script` that is archived in the directory `condir/`. An example of an archived file is shown in Figure 12.

## 1.8 Geometry files

Most of the detector geometry for both generation (program file `source/fastmc.f`) and analysis (program file `source/anal.f`) is provided by a single geometry file. It is the user's responsibility to correctly associate the same geometry file for both generation and analysis. Some geometry is hard-coded into the analysis job such as the location

Key words	Format	Description
DEBUG	I	0 produces no debug. > 0 produces increasing debug output
EVENT SOURCE	I	0 = generate events, 1 = read from file (see HFIN)
COPY JOB	I	0=not a copy job, otherwise yes
EVENTS	I	Number of events to generate
SEEDS	III	seed1 seed2 nskip1 nskip2 Two random number seeds and number of random numbers to skip
NEUTRON BEAM	IRRRR	N,P,x,y,z N≠0, then implement neutrons. N = 2, then use measured neutron spectrum (P is irrelevant) P = neutron momentum. x,y,z = interaction point
KBEAM	IRR	N, P, $\sigma(P)$ (P, $\sigma(P)$ irrelevant unless N = 0) N = 0 : gaussian of mean P, width $\sigma(P)$ N ≥ 1 : spectrum from <code>source/kspec.f</code> N = -1 : spectrum from <code>source/kapinos.f</code> (no angular dependence, assume 45° production angle) N ≤ -2 : spectrum from <code>source/kapinos2.f</code> (angular dependence)  N  > 1 and N ≠ -2 : generate <code>AveK0LperBucket</code> $K_L^0$ /event <code>AveK0LperBucket</code> = 3.57, defined in <code>source/detector.inc</code>
COLLIMATION	I	≠ 0 then use BEAM_COLLIMATION (see <code>source/beam_collimator.f</code> )
EXTENDED_TARGET	I	≠ 0 then use extended target (see <code>source/gen_vertex.v07_2.f</code> )
K0Ldecaymode	I	Select $K_L^0$ decay mode K. (see Table 4) K = 0 selects randomly amongst 14 currently allowed modes according to branching fraction. 1 ≤ K ≤ 20 selects a currently defined decay modes. K > 20 generates an error.
ANNIHILATION	I	0 = no $e^+$ annihilation 1 = true $e^+$ annihilation (see <code>source/ganni.f</code> ) 2 = approximate $e^+$ annihilation ( $P(\gamma) = P(e^+)$ )
CONVERSION POINT	I	Select $Z_\gamma$ = photon conversion point and PreRadiator model 0: $Z_\gamma$ = Zdeath (usually rear of CAL), Konaka PR model 1: $Z_\gamma$ = middle of PR, Konaka PR model 2: $Z_\gamma$ = front of PR, Zeller PR model 3: $Z_\gamma$ = rear of PR, Konaka PR model 4: $Z_\gamma$ = front of PR, Bryman PR model 5: $Z_\gamma$ = rear of PR, Bryman PR model
USE K WEIGHT	IRR	V, $\delta Z_1, \delta Z_2$ V ≠ 0 enable weighting of $K_L^0$ decays Generate $Z(K_L^0)$ using $Z_1 - \delta Z_1 \leq Z(K_L^0) \leq Z_2 + \delta Z_2$ $Z_{1(2)} = \text{US(DS)}$ end of decay region

Table 2: Part 1 of key words, format and description used in control file. Format of “I” means integer, “R” means real and “C” means character string. All variables **must** be given even if the values are irrelevant.

Key words	Format	Description
PION BETA DECAY	I	If $\neq 0$ , set $\mathcal{B}(\pi^\pm \rightarrow \pi^0 e^\pm \nu) = 1$
MAXIMUM COS(K,e) FOR KE4	R	Maximum $\cos(K_L^0, e)$ for Ke4 decay (No cut if $> 1$ obviously)
SMEARING	I	0 = no smearing 2 = smear as PR conversion 1 = smear as CAL conversion( <b>NOT IMPLEMENTED</b> )
PI0 FIT	I	if $\neq 0$ then do constrained fit to $\pi^0$ candidates
WHICH FITS?	III	Which fits to do? 0=no, otherwise yes Order is $2\gamma, 1\gamma, 0\gamma$ conversions in PR
FORCE YBEAM HEIGHT IN FIT	R	$V$ = vertical beam size used in $\pi^0$ fit if $V \leq 0$ , use nominal Y beam angle to compute beam size, nominal beam angle from BEAM DIVERGENCE, Figure 15
TRIGGER	III	$i_1 i_2 j_1 j_2$ for 'trigger' requirement $i_{1(2)}$ = minimum(maximum) number of hits in PR/CAL $j_{1(2)}$ = min(max) number of hits in decay volume veto region -1 1000 -1 100 will accept all events
SKIM ACTIVATE	I	0 = no skim, otherwise do skim
SKIM PHOTON	RRRR	Xmax Ymax Xmin Ymin : skim requirement on photon position Xmax,Ymax are maximum $ X_\gamma ,  Y_\gamma $ at 10 $X_0$ into CAL Xmin,Ymin are minimum $ X_\gamma ,  Y_\gamma $ at front of CAL
SKIM KAON	RRRR	Emin Emax Zmin Zmax : skim requirements on FITTED Energy, Z-position of $K_L^0$
SKIM PION	RRRRR	$M_{\gamma\gamma, \min} M_{\gamma\gamma, \max} E_{\min}^* E_{\max}^* s_{\min}$ : skim requirements on Fitted $M_{\gamma\gamma}, E^*(\pi^0), s$ where $s \equiv E^*(\pi^0) -  E_{\gamma 1}^* - E_{\gamma 2}^* $
SKIM KP2	I	= 0,1,2 = all decays, Kp2 Odd only, Kp2 Even Only accepted
NTUPLE TYPE	I	= 0,1,2,3 = none,jaffe-style,konaka-style,both styles jaffe-style and konaka-style ntuples are put in different output files
HFIN	CR	Input ntuple file name and time offset in ns Multiple HFIN cards will be read and used in order
HFNAME	C	output histogram file name (maximum of 132 characters) if multiple lines with HFNAME, use them in order
GZIP OUTPUT NTUPLE	I	= 0 no gzip, otherwise gzip
FAKE AND MERGED PHOTONS	IIIRRR	STOPPED_MUONS NEUTRON_SHOWERS MERGE_PHOTONS Rmoliere Nmoliere dTmerge STOPPED_MUONS = $(i_{BV} i_{OV} i_{CAL} i_{PR})$ NEUTRON_SHOWERS = $(j_{BV} j_{OV} j_{CAL} j_{PR})$ if $k_{SS} \neq 0$ then implement fake photons from stopped muons( $k = i$ ) or neutron showers( $k = j$ ) for subsystem SS = BV,OV,CAL,PR MERGE_PHOTONS = 0 = no merge, otherwise yes Photons are merged if they are within Nmoliere Molière radii Rmoliere(cm) and within dTmerge(ns) in time

Table 3: Part 2 of key words, format and description used in control file. Format of “I” means integer, “R” means real and “C” means character string. All variables **must** be given even if the values are irrelevant. “SKIM” requirements are ignored if skimming is not activated. The key words “MAXIMUM COS(K,e) FOR KE4”, “FORCE YBEAM HEIGHT IN FIT” and others are written in a small font so that I can cram them into the columns of this table.

Mode number	Name	Final state	Branching fraction	$\mathcal{B}/\mathcal{B}(K_L^0 \rightarrow \pi^0 \nu \bar{\nu})$
1	kpnn	$\pi^0 \nu \bar{\nu}$	$0.3000 \times 10^{-10}$	1.000
2	kp2	$\pi^0 \pi^0$	$0.9270 \times 10^{-03}$	$0.3090 \times 10^{+08}$
3	kcp2	$\pi^+ \pi^-$	$0.2056 \times 10^{-02}$	$0.6853 \times 10^{+08}$
4	kgg	$\gamma \gamma$	$0.5860 \times 10^{-03}$	$0.1953 \times 10^{+08}$
5	kp3	$\pi^0 \pi^0 \pi^0$	0.2113	$0.7043 \times 10^{+10}$
6	kcp3	$\pi^+ \pi^- \pi^0$	0.1255	$0.4183 \times 10^{+10}$
7	ke3	$\pi^\pm e^\mp \nu$	0.3878	$0.1293 \times 10^{+11}$
8	km3	$\pi^\pm \mu^\mp \nu$	0.2718	$0.9060 \times 10^{+10}$
9	ke3g	$\pi^\pm e^\mp \nu \gamma$	$0.3620 \times 10^{-02}$	$0.1207 \times 10^{+09}$
10	km3g	$\pi^\pm \mu^\mp \nu \gamma$	$0.5700 \times 10^{-03}$	$0.1900 \times 10^{+08}$
11	kpgg	$\pi^0 \gamma \gamma$	$0.1680 \times 10^{-05}$	$0.5600 \times 10^{+05}$
12	ke4	$\pi^0 \pi^\pm e^\mp \nu$	$0.5180 \times 10^{-04}$	$0.1727 \times 10^{+07}$
13	km4	$\pi^0 \pi^\pm \mu^\mp \nu$	$0.1400 \times 10^{-04}$	$0.4667 \times 10^{+06}$
14	ke2g	$e^+ e^- \gamma$	$1.0 \times 10^{-5}$	
15	km2g	$\mu^+ \mu^- \gamma$	$3.59 \times 10^{-7}$	
16	kpee	$\pi^0 e^+ e^-$	$5.1 \times 10^{-10}$	
17	kpmm	$\pi^0 \mu^+ \mu^-$	$3.8 \times 10^{-10}$	
18	kgnn	$\gamma \nu \bar{\nu}$	$2.0 \times 10^{-13}$	
19	kbet	$K^\pm e^\mp \nu$	$0.1000 \times 10^{-07}$	333.3
20	knn	$\nu \bar{\nu}$	$1.0 \times 10^{-20}$	

Mode number		Decay	Branching fraction	ME or PS?
1	-	$\pi^0 \rightarrow \gamma \gamma$	0.98798	-
2	-	$\pi^0 \rightarrow \gamma e^+ e^-$	0.01202	ME
1	-	$\pi^\pm \rightarrow \mu^\pm \nu$	0.999877	-
2	-	$\pi^\pm \rightarrow e^\pm \nu$	$1 - 0.999877$	-
3	-	$\pi^\pm \rightarrow \pi^0 e^\pm \nu$	$1.025 \times 10^{-8}$	PS
1	-	$\mu^\pm \rightarrow e^\pm \nu \bar{\nu}$	0.986	ME
2	-	$\mu^\pm \rightarrow e^\pm \nu \bar{\nu} \gamma$	0.014	ME
1	-	$K^\pm \rightarrow \pi^\pm \pi^0$	1	-

Table 4: The upper part of the table lists the number, four-character names of the allowed  $K_L^0$  decay modes in the FastMC. A reasonable approximation to the correct matrix element is used for all  $K_L^0$  decays except for  $K_L^0 \rightarrow K^\pm e^\mp \nu$  which uses phase space. The lower half lists the other particle decays, mode numbers and branching fractions. The rightmost column indicates whether the matrix element (ME) or phase space (PS) is used for the decay dynamics.

of the catcher and vetos around the upstream beam hole. The most recent geometry file is reproduced in Figure 13,14,15.

## 1.9 Production of ntuples

Two types of ntuples can be produced by `fastmc.exe`, these are called Akira-style and Jaffe-style for historical reasons. Akira-style lends itself to paw-based analysis using `kumacs` and functions in `analysis/` directory. I don't use these, so I've not modified them. Jaffe-style lends itself to fortran-based analysis although paw-based analysis is also possible, of course.

A ntuple of one type can be quickly converted to the other using a 'copy job' option.

A single job can produce both types of ntuple simultaneously.

It is also possible to 'skim' events with some loose selection cuts. Skim statistics are maintained in `DataBase/skim.statistics.and.log` and read in for later analysis (only available for Jaffe-style ntuples with fortran-based analysis).

Ntuples can also be read in with different time offsets for multi-bunch studies (HFIN in Figure 3).

## 2 Track propagation and decay weights

The detector geometry is defined as a series of rectangular parallelepiped volumes. Each particle is tracked from its origin to the boundary of the current volume or until its decay or interaction point. If the boundary is contiguous with an adjacent volume, then tracking of the particle continues. Tracking of a particle terminates with decay, interaction or intersection with the boundary of the most upstream or most downstream volume for upstream-going or downstream-going particles, respectively. If a decay or interaction is forced, then the decay or interaction probability is stored in the per-particle weight (See `Weight` in Figure 16). Currently only  $K_L^0$  decays and  $e^+$  annihilations receive a per-particle weight different from unity and are assigned in `source/fastmc.f`. The  $e^+$  annihilation may be re-calculated in `analysis.script` to use the charge veto inefficiency function, see Figure 7 and `source/gettrkweight.f`.

## 3 The stack and `source/trackinfo.inc`

Created particles, such as  $K_L^0$  or  $n$  are placed on the stack (`source/evolve.f`). When a particle trajectory is terminated, either by striking a boundary, decaying or interacting, any daughter particles are placed on the stack. The last particle on the stack is processed first until there are no particles remaining on the stack.

When a particle is placed on the stack, its initial position, time and four-momentum are stored in `xBirth,yBirth,zBirth,Tbirth` and `PxBirth,PyBirth,PzBirth,Ebirth` in `source/trackinfo.inc` (Figure 16). The four-momentum in the  $K_L^0$  rest frame `PxinkCMS,PyinkCMS,PzinkCMS,EinkCMS` is also stored, if applicable, in `source/fastmc.f`. The type of particle `Ipart` (Table 5), its electric charge `Charge` and the track number of particle's parent `Parent` is also stored. If a particle has no parent (for  $K_L^0$  and  $n$ , for

Number	Name
1	photon
2	neutrino
3	electron
4	muon
5	neutral pion
6	charged pion
7	charged kaon
8	$K_L^0$
9	$K_S^0$
10	proton
11	neutron
12	fake photon from stopped muon
13	fake photon from neutron-induced shower
14	merged photon

Table 5: Particle number and type for particles in the FastMC. A “merged photon” can be created from a pair of real or fake photons.

example, then `Parent=0`). See Section 4.3 for the ‘parents’ of merged photons. The particle type `Ipart` is defined in `source/partprop.inc`.

When a particle trajectory is terminated, the final position and four-momentum is stored in `xDeath,yDeath,zDeath` and `PxDeath,PyDeath,PzDeath,EDeath`, respectively. Note that these quantities can be over-written for photons that are “smeared” to approximate resolution effects (Section 8). In addition the `Fate` of each particle is stored, the allowed values of `Fate` are defined in `source/detector.inc` as parameters `Kdecayed = 1`, `Khit = 2`, `KAnni = 3` or `Kmrg = 4` for particles that have decayed, hit a boundary, annihilated, or been merged into ‘merged photon’ respectively. The final volume number `LastVol` and plane in the final volume `LastPL` are also stored.

## 4 Fake and merged photons

Three processes that may affect signal or background yields are

1. Showers from decays of muons that have stopped in the detector,
2. Showers induced by neutron interactions, and
3. Merging of nearby showers into a single shower.

Showers from the first two processes are referred to as “fake” photons in the FastMC. Both real and “fake” photons can be merged into a single “merged photon”. The implementation and assumptions for fake and merged photons are described below.

## 4.1 Fake photons from stopped muons

The subdetector, time  $t_\mu$ , position  $\vec{x}_\mu$  and three-momentum of  $e^\pm$  from stopped, decaying muons produced in  $K_L^0$  decays identified in the KOPIO GEANT Monte Carlo were extracted and written to a file for the BV, OV, CAL and PR subdetectors. The mean number of stopped decaying muons per microbunch  $\mu_\mu$  was determined from the same sample of simulated events. (The GEANT simulation assumed that 3.57 is the mean number of  $K_L^0$  exiting the spoiler per microbunch.) When STOPPED\_MUONS are enabled in the control file (Table 3),  $N_\mu$  stopped muons are selected from the file where  $N_\mu$  is drawn from a Poisson distribution of mean  $\mu_\mu$ . For stopped muons in the PR, only those  $e^\pm$  with a positive Z component of momentum are put only the stack. It is assumed that we can identify the direction of an electromagnetic shower in the PR and reject backward-going showers. Stopped muons in other subdetectors are loaded directly into the stack. The time  $t$  of the muon on the stack is  $t = \text{mod}(t_\mu - t_0, \Delta T) + t_0$  where  $t_0 = d/c$  and  $d$  is the distance between  $\vec{x}_\mu$  and the origin and  $\Delta T = 40$  ns. For convenience the initial “birth” and “death” four-momenta, positions and times of fake photons are taken to be the same. The “death” parameters may be subsequently changed by smearing.

## 4.2 Fake photons from neutron-induced showers

A sample of neutron-induced showers were simulated using GEANT with a PR and CAL detector configuration similar to that used by the KOPIO GEANT Monte Carlo. The BV and OV were not simulated. Simple clustering algorithms were used to associate PR or CAL elements and create ‘reconstructed showers’. For showers in the PR, the position and time of energy deposit in the first and second elements of the shower are used to establish the direction of the shower. As for stopped muons, only forward-going showers in the PR are accepted.

Several adjustments are made to GEANT information to conform to the FastMC geometry and to correct the shower energy. The shower energies in the PR and CAL are corrected by factors of 1.12 and 1.24, respectively. These factors were determined from samples of simulated 100 MeV and 250 MeV photons reconstructed with the same algorithms.

The time of the neutron-induced shower on the stack is calculated in a similar fashion to that of stopped muons and the same comments for the “birth” and “death” parameters apply.

The mean number of neutron-induced showers per microbunch  $\mu_n = 0.672$  is determined by

$$\mu_n = n/p/\mu\text{sr} \times A \times t_{\text{spill}} \times f_{\text{RF}} \times a_{\text{spoiler}} \times f_{\text{target}} \times P_{\text{shower}}$$

where

- $n/p/\mu\text{sr} = 1.66 \times 10^{-6}$  neutrons per incident proton per  $\mu\text{sr}$  for neutrons above 10 MeV from TN059, section 6 from 1999 proposal, Table 4, p. 142.
- $A = 100 \times 5 \mu\text{sr}$  is the aspect ratio of the neutral beam,

- $t_{\text{spill}} = 2.4 \text{ s}$  is the length of the spill,
- $f_{\text{RF}} = 25 \text{ MHz}$  is the RF frequency,
- $a_{\text{spoiler}} = 0.5$  is the spoiler attenuation factor according to Andrei Poblaguev,
- $f_{\text{target}} = 0.85$  is the reduction due to a non-pointlike target according to Andrei Poblaguev, and
- $P_{\text{shower}}$  is the probability per beam neutron of producing a shower.

$P_{\text{shower}}$  is determined from the same GEANT simulation for 40 million beam neutrons to be  $P_{\text{shower}} = (4311 \times 10 + 1824 \times 1 + 804 \times 1)/40 \times 10^6 = 1.143 \times 10^{-3}$ . The 40 million beam neutrons were divided into three samples based upon position and trajectory at the end of the collimation system. Events from the first (second and third) sample(s) were used once (ten times) for further propagation into the detector; hence the weighting in the expression above.

### 4.3 Merging of photons

Pairs of photons, both real and fake, can be merged if they are within Nmolierie Molière radii  $R_M$  of each other at the front of the CAL, BV or OV. The Molière radius of the CAL is 5.98 cm (V.Issakov for modules with 0.275 mm Pb and 1.5 mm scintillator layers) and is used for the  $R_M$  of the BV and CAL. Two values of Nmolierie of 5 and 7 are proposed as “optimistic” and “pessimistic”, respectively, based on the experience of E865 [9]. The E865 calorimeter had shashlyk modules with transverse size  $11.4 \times 11.4 \text{ cm}^2$  which corresponded to  $2.5 R_M$ . Simulation of the E865 calorimeter showed that a 1 GeV photon deposited 99% (95%) of its energy in a  $3 \times 3$  ( $2 \times 2$ ) array of modules; hence, it is reasonable to assume that two photons separated by  $7 R_M$  could definitely be distinguished from a single photon. The lateral shower size of two photons separated by  $5 R_M$  might be distinguished from that of a single photon. It should be kept in mind that the typical energy of a photon in KOPIO is 250 MeV and not 1 GeV as in E865.

Pairs of photons for merging are also required to be within dTmerge ns of each other. According to Mike Zeller, the double-pulse resolution of the CAL with APDs is 15 ns, so the default value of dTmerge is 15 ns.

Note that the Nmolierie, Rmolierie and dTmerge are user-adjustable parameters and can be changed if the above arguments prove to be incorrect.

Only pairs of photons are considered for merging; that is, if three or more photons satisfy the merging criteria above only the nearest two photons are merged. This assumes that the showers of multiple photons would always be distinguishable from that of a single photon.

The four-momentum of a merged photon is taken as the sum of the four-momentum of the two photons. The position is taken as the energy-weighted average of the two photons. The two photons that have been merged are assigned a 'Fate' of `Kmrg = 4` to distinguish them from photons that have not been merged. In addition the 'parent' of a merged photon is `parent = -(i_1 + 1000 \times i_2)` where  $i_1$  and  $i_2$  are the track numbers of

the two photons. The “birth” and “death” parameters are assigned in a similar fashion as for the fake photons.

## 5 Neutron propagation and interaction

Neutrons are propagated from an origin in the target at  $\vec{x} = (0, 0, 0)$  to a fixed interaction point specified by the user (see NEUTRON BEAM in Table 2). The  $nN \rightarrow \pi^0 NX$  interaction for  $n$  above  $\pi^0$  production threshold is simulated assuming phase space and baryon number conservation for the final state. The neutron production spectrum is shown in Figure 8 and is taken from the measured neutron energy spectrum at  $46.5^\circ$  from Figure 18 of the TDR [1]. Halo neutrons are assumed to have the same momentum spectrum. There is no angular dependence to the neutron production spectrum.

For rate estimates, the  $n/K_L^0$  production ratio is assumed to be 1000 and the rate of halo to beam neutrons taken as 0.0001. Both these values are hard-coded in file `source/neutronnorm.f`.

## 6 $K_L^0$ spectrum and production rate

There are several possible choices for the  $K_L^0$  production spectrum. The user can specify the mean and standard deviation of a gaussian spectrum with no angular dependence, or a  $K_L^0$  spectrum based on Kapinos [2] with and without angular dependence. The angular dependence is approximated by interpolation or extrapolation using measurements at production angles of  $40^\circ$  and  $45^\circ$ .

The production rate is calculated based on 3.57  $K_L^0$  per microbunch, a 12 cm long platinum target (`DataBase/primary_beam.dat`) with a secondary production rate of 0.79 [3], 12000 hours of beam with a 2.4 second spill and 2.3 second interspill period. The reduction due to the requirement of a single  $K_L^0$  per microbunch is taken to be 0.646 (all these values are hard-coded in `source/anal.f`).

## 7 $K_L^0$ Production point and beam aspect ratio

The user can optionally specify a point target at  $\vec{x} = (0, 0, 0)$  or an extended target and incident proton beam as specified in the file `DataBase/primary_beam.dat`. The current specification for the extended target and beam is taken directly from the KOPIO GEANT MC and the KOPIO parameters [4]. The  $K_L^0$  production point in the extended target assumes exponential attenuation of the incident proton beam.  $K_L^0$  production from secondaries is ignored in the generation of the  $K_L^0$  production point in the extended target.

The user can specify the beam aspect ratio and supplement it with a collimator system taken from the KOPIO GEANT MC which is based upon Geometry V [5]. Limits to the beam in X and Y at 10 different Z positions are calculated in the KOPIO GEANT MC and used in the FastMC. The limits are in file `geometry/collimator.v07_4.dat`. The collimation system is only applicable for the  $100 \times 5$  mrad<sup>2</sup> aspect ratio. When the

collimator system is enabled, the allowed angular range of  $K_L^0$  production is increased so that the collimation system determines the accepted aspect ratio.

## 8 Smearing of hits

The smearing of photon hits is performed in `source/smear.f`. “Good” photons are propagated to the nominal conversion point selected by `CONVERSION POINT` as described in Table 2. “Good” photons are defined in `source/goodphoton.f` as photons above an energy threshold (`EPRthres` nominally 10 MeV, set in `source/detector.f`) that were capable of reaching the nominal conversion point. The energy and time are smeared assuming gaussian resolution functions according the parameters given in the geometry file (Figure 15). The x and y positions and angles are smeared according to the option selected by `CONVERSION POINT`. The z position is unsmeared.

For the Konaka PR model, a double-gaussian approximation to the resolution is used and the position and angular smearing are given by parameters in the geometry file. For the Bryman PR model, the angular resolution is parametrized as a double-gaussian with widths that are a function of energy and angle as described in [6] and implemented in `source/get_pr_resolution.f`. For both the Konaka and Bryman PR models, the resolutions used in the constrained fit are taken from the narrow gaussian of the models.

The Zeller PR model (`source/prerad.f`) is “A program to reconstruct the response of analog strip pre-radiator for E926. The basic geometry is 2 nested hexagonal tubes with readout between, separated by Pb radiator of thickness TPb (cm). x=0, y=0 is beam line, z0(I)=radiator point. Takes average of two analog hits at each plane. Randomize which comes first, x or y.” It uses pair production and bremsstrahlung based on GEANT. The resolutions used in the constrained fit use hard-coded, empirically energy-dependent resolution for the Zeller PR model.

The smeared quantities replace the stored quantities `PxDeath`, `PyDeath`, `PzDeath`, `EDeath` for the four-momentum and `xDeath`, `yDeath`, `zDeath` for the final position of each photon. In addition the variable `smeared` indicates how a photon was smeared (see Figure 16).

### 8.1 Contrasting Bryman and Zeller PR models

(This section is taken from an e-mail from Doug Bryman, Jun 11 2004.)

I finally looked more closely into the “Zeller” model of prerad reconstruction in comparison with the “Bryman” model and here’s what I conclude:

Mike used only the analog/strip readout model for the chambers and parameterized the angular dependence of the resolutions for x and y as  $G_{meas} = \sigma_0 + 0.1 * \tan(\theta)$ . This is about 25% worse than actually measured for strips only in the TN106 [7] test (fig. 15). [In the Bryman model, I used a slightly better approximation to the TN016 [7] data  $G_{meas} = \sqrt{\sigma_0^2 + \sigma_1^2 * \tan^2(\theta)}$  and in the June 9, 2003 addendum to TN014 [6] reproduced the TN016 [7] angular resolution for strips only reasonably well (fig. A4). I also reproduced the TN016 [7] anode only angular resolution data

fairly well.]

Other features of the Zeller model:

- 95% efficiency per measurement.
- MCS selected from a Gaussian distribution.
- x and y slopes calculated separately.

The largest difference between "Zeller" and "Bryman" is that I used a model of both wire readout (drift time) and strip readout with alternating x and y wires (strips). This drastically reduced the angular dependence of the angular resolution as indicated in TN014 [6] figs. A4 and A8 mainly because the wire resolution is insensitive to angle (although the best resolution for wires is worse than for strips). This was embedded in a "toy" preradiator Geant calculation (including accurate MCS distributions, bremsstrahlung, etc.). The angular resolution was determined simultaneously using both x and y measured points and the resolution distributions were fit with 2 Gaussians for parameterizing.

**So, my conclusion is that we should use the Bryman model as the basis for current estimates or the Zeller model should be revised to take into account the alternating wire/strip directions and include a better approximation to the position resolutions.** From a preliminary running of the fastmc with the Bryman model I get more events with an improved S/N than with the Zeller model. This is obviously to be expected since the effective angular resolution is quite a bit better and has little dependence on angle.

## 9 Reconstruction and fitting

All pairs of "good" photons are considered as possible  $\pi^0$  candidates by `source/recon.f`. Up to `Mrecon` (=50 in `source/reconinfo.inc`, Figures 17,18, and 19) reconstructed pairs are allowed per event, if more are found, the event is discarded (this only happens when many  $K_L^0$  decay near the signal region). Two sequential kinematic fits are attempted for each candidate pair (`source/fitpi0.f`). The first fit does not impose the  $\pi^0$  mass constraint, and the second fit does impose the mass constraint. Photon pairs giving ludicrous fit results, defined as  $|v_i(K_L^0)| > 10^6$  cm or ns for  $v_i = x, y, z, t$  of the  $K_L^0$  candidate, are discarded. If the reconstructed production vertex of candidate  $\pi^0$  (and  $K_L^0$ ) produces a physically meaningful value of  $\beta$  when the  $K_L^0$  production point is assumed to be  $\vec{x} = (0, 0, 0)$  cm at time  $t = 0$  ns, then the ntuple variable `RecValidKL` is set true. **Users should require `RecValidKL` to be true when doing analysis.** For each  $\pi^0$  candidate passing these basic criteria, the reconstructed quantities listed and described in Figure 17 are stored.

Ntuple variables giving the calculated uncertainty of some variables:

- `RecSMass` = uncertainty on `RecMass`, the two photon mass,
- `RecCovX4Kf1` = packed covariance matrix of the fitted  $K_L^0$  decay position and time 4-vector  $(x, y, z, t)$  from the first fit,

- `RecCovX4K` = same as `RecCovX4Kf1` but for the second fit,
- `RecCov1` = packed covariance matrix for the following four variables:
  1.  $E_\pi^*$ ,
  2.  $\Delta E^* = E_1^* - E_2^*$ ,
  3.  $E_{\text{miss}}$ , and
  4.  $\sqrt{E_\pi^{*2} - \Delta E^{*2}}$ .

Covariance matrices are packed using the following method:

1. The covariance matrix  $\mathcal{C}$  is transformed into the correlation matrix  $\mathcal{R}$  with  $\mathcal{R}_{ij} = \mathcal{C}_{ij}/\sqrt{\mathcal{C}_{ii}\mathcal{C}_{jj}}$ , but since the diagonal elements of  $\mathcal{R}$  are unity by definition, they are replaced by  $\sqrt{\mathcal{C}_{ii}}$ .
2.  $\mathcal{R}$  is packed so only the lower-left part is saved in a linear array in the following order  $\mathcal{R}_{11}, \mathcal{R}_{21}, \mathcal{R}_{22}, \mathcal{R}_{31}, \mathcal{R}_{32}, \mathcal{R}_{33}, \mathcal{R}_{41}, \mathcal{R}_{42}, \mathcal{R}_{43}, \mathcal{R}_{44}$ .

Currently it is possible to fit each photon pair as if

1. Both photons converted in PR (`TypeOfFit = 0`),
2. One photon converted in PR, one photon converted in CAL (`TypeOfFit = 1`) and/or
3. Both photons converted in CAL (`TypeOfFit = 2`).

This means that there can be up to three reconstructed  $\pi^0$  candidates for each photon pair depending on WHICH FITS? in the control file Figure 11. The different fits are differentiated during analysis by `TypeOfFit`. If the photon is assumed to have converted in the CAL, then the angular information is ignored in the fit in order to approximate the worse resolution of the CAL. Only the first two of these options can be analyzed with `analysis.script` because no cuts have been defined for the third option (see variable `AllowedFits` in `source/setupanal.f`).

## 10 Changing the radiation lengths of the PR

The radiation length of the PR is currently hard-coded as  $2X_0$ .

During generation `run.fastmc.script` for the Zeller model, one must change the variable `rad_len` in `source/prerad.f`.

During analysis `analysis.script`, one must change `RadLenPR` in `source/setupanal.f`.

## 11 How yields are calculated in `analysis.script`

The expected number of events and the corresponding uncertainty are reported by the `analysis.script`. For a given decay mode ( $i$ ) and a given set of cuts ( $l$ ), the yield  $Y_{il}$  is

$$Y_{il} = A_{il} \mathcal{B}_i N_K / N_i \quad (2)$$

where

- $A_{il} = \sum_k^{n_i} w_{ik}$ ,
- $w_{ik}$  = weight of  $k^{\text{th}}$  candidate from mode  $i$ ,
- $n_i$  = number of candidates for mode  $i$ ,
- $\mathcal{B}_i$  is the  $K_L^0$  branching fraction for mode  $i$  (see Table 4),
- $N_K$  = is the total number of  $K_L^0$  exiting the spoiler (see Section 6) and
- $N_i$  = is the number of generated events for mode  $i$ .

Note that there can be multiple  $\pi^0$  candidates in a single event. For example in a  $K_L^0 \rightarrow \pi^0 \pi^0$  decay, if three of the four photons satisfy the fiducial requirements in the PR and CAL, then there can be up to three candidates from the pairings  $(\gamma_1 \gamma_2)$ ,  $(\gamma_1 \gamma_3)$  or  $(\gamma_2 \gamma_3)$ .

Sets of cuts are defined and made in `source/select_signal.f` and `source/akira.offline.f`. For historical reasons there is one cut set labelled “Zcuts” or “MZ”, derived from cuts proposed by Mike Zeller and a series of increasingly tighter cuts labelled “AK pre-basic”, “AK basic”, “AK loose”, “AK lominal”, “AK tight”, “AK tighter” and “AK tightest” designed by Akira Konaka and Francesca Marcucci [8].

The weight  $w_{ik}$  for a candidate is a product of the relevant individual track weights and is calculated in `source/getcandwt.f` and `source/gettrkweight.f` (Section 12).

$$w = \prod_j^{N_t} d_j v_j c_j l_j \quad (3)$$

where the subscripts for  $w_{ik}$  have been suppressed and

- $j$  is the track number,
- $N_t$  is the number of tracks in the event,
- $d_j$  is the decay weight associated with track  $j$  (see Section 2),
- $v_j$  is the veto inefficiency of track  $j$ ,
- $c_j$  is the conversion probability of track  $j$  and
- $l_j$  is the loss due to attenuation of photons in the vacuum vessel for track  $j$ .

For the two photons considered as daughters of the candidate  $\pi^0$ ,  $v_j \equiv 1$ ,  $c_j \leq 1$  depending on the path length in the PR and whether the photon is considered as converting in the PR or in the CAL, and  $l_j = 1$  unless the variable `AllowVVloss` is set `.TRUE.` in file `source/setupanal.f`. The probability of conversion for path length  $L$  is taken as  $1 - e^{-7/9L/R}$  where  $R$  is the radiation length of the PR. The calculated path length takes into account the entrance and exit point of the photon trajectory in the PR (`source/getpath.f`). Photons are allowed to enter the PR from the beam hole. The loss of photons due to attenuation in the beam pipe and other material is currently not taken into account.

For photons not considered as daughters of the candidate  $\pi^0$ ,  $v_j$  is calculated based on the photon trajectory and energy, and  $c_j = l_j = 1$ . The photon veto inefficiency is shown for three possible options in Figure 3. The functional form of the standard PV function is given in Figure 4. The different options can be selected in `source/setupanal.f`. For photons that hit the catcher, the standard veto inefficiency is taken to be 0.01. Various options are available for the catcher veto inefficiency in `source/setupanal.f`.

For charged particles in the final state (that is, those which have not annihilated or decayed),  $v_j$  is calculated based on the particle trajectory, species and energy. See Figure 7 and `source/chgveto.f` (Figures 5 and 6).

If a charged particle or photon passes through the upstream beam hole, then  $v_j = 1$ .

Note that the particle energy used to calculate the veto inefficiency is the initial energy (`Ebirth` (Section 3) and not the smeared energy (`Edeath` (Section 8)).

Timing is currently ignored in the assignment of  $v_j$ .

## 12 More details of candidate weight calculation

This section describes how the weight of a single candidate (Eqn. 3) is calculated in `source/getcandwt.f`.

First a list of the low energy photons produced by  $e^+$  annihilation that will not be considered for photon vetoing is compiled depending on variables `Neglect2dG`, `RequireSep2dG` and `Cos2g`. The default is to consider the low energy annihilation photon if the opening angle between the photons is  $> 20^\circ$ .

For each charged track or photon the decay weight  $d_j$  (Section 2), veto inefficiency  $v_j$ , conversion probability  $c_j$  and attenuation loss  $l_j$  is calculated in `source/gettrkweight.f`. In the following “track” refers to either charged tracks or photons.

Tracks that exited the decay volume through the US beam hole are flagged as well as photons that hit the catcher. The size of the US hole and catcher is set by the geometry file (Figure 13). The default photon veto inefficiency of the catcher is 1%.

The veto inefficiency of tracks in the US hole is calculated in `source/usveto.f` and assumes that the veto inefficiency is adjusted for the conversion probability ( $v_j = f(E_\gamma)/c(s)$  where  $f(E_\gamma)$  is the function shown in Figure 3 and  $c(s)$  is the conversion probability given a path length  $s$  in the US veto. **There is an error for FastMC versions v01\_1 and earlier with  $v_j = f(E_\gamma) \times c(s)$ .** Charged tracks that traverse the beam pipe (0.6 cm thick) and  $> 0.5$  cm of the US veto are assigned the charged

Z	X min	Y min	X max	Y max	Fiducial limits/sizes	SETUPANAL
1015.0	-1.0	-1.0	65.0	5.0	US hole	
1415.0	-1.0	-1.0	110.2	11.6	DS hole	
1415.0	-1.0	-1.0	150.0	150.0	PR front	
1515.0	110.2	11.6	200.0	200.0	CAL front	
1557.7	-1.0	-1.0	250.0	250.0	CAL 10r1	
1595.0	-1.0	-1.0	250.0	250.0	CAL rear	
2715.0	-1.0	-1.0	200.0	10.0	Catcher front	

Figure 2: The  $|X|$  and  $|Y|$  fiducial limits on candidate photons. A value less than zero means no limit is applied.

veto inefficiency shown in Figure 7 depending on their species except for positrons which are assigned the  $\mu^+$  veto inefficiency to somehow account for the probability of detecting the shower.

The veto inefficiency of all other tracks is energy- and species-dependent (but not position-dependent) and is shown in Figures 4, 5 and 6 for photons and charged tracks, respectively. This is the explicit assumption of hermiticity and the neglect of timing information.

The conversion probability in the PR  $c_j^{\text{PR}}$  is calculated based on the path length  $s$  of the photon trajectory in the PR,  $c_j^{\text{PR}} = 1 - \exp(-7/9 \times s/X_0)$  where  $X_0$  is the number of radiation lengths of the PR. The conversion probability for a photon in the CAL is  $c_j^{\text{CAL}} = 1 - c_j^{\text{PR}}$ .

The geometric acceptance of the candidate photons is determined in `source/acceptable.f`. The fiducial limits are set by the specifications in the geometry file and reported in the analysis job (Figure 2).

## 13 Possible improvements/modifications

The following improvements/modifications could be done at generation.

1. Define resolution for CAL, BV or OV hits.

The following improvements could be done in the analysis phase.

1. Analysis “control file” so that the analysis program does not have to be re-compiled for every change.
2. Inclusion of D4 magnetic field.
3. Define analysis cuts for two photon conversions in CAL.
4. Define analysis cuts for photon pairs consisting of one PR conversion and one conversion in the BV or OV.
5. Position-dependent photon and/or charged veto inefficiency.

6. Account for losses due to material in front of the PR other than the vacuum vessel.

## 14 Acknowledgements

Marvin Blecher and Andrei Poblaguev produced the stopped muon and neutron-induced shower files. Vladimir Issakov, Mike Zeller and Andrei Poblaguev provided guidance on the response of the calorimeter. Dima Vavilov wrote the code to calculate the uncertainties on fitted quantities. Other KOPIOs provided feedback and suggestions, sorry if you are not mentioned.

## References

- [1] “KOPIO: Measurement of the decay  $K_L \rightarrow \pi^0 \nu \bar{\nu}$ , Draft Technical Design Report for the National Science Foundation, TDR, 8 June 2001.
- [2] P. Kapinos, ”The estimation of Klong flux at 24 Gev/c”, KOPIO Technical Note 5, 22 Aug 1997.
- [3] L.Littenberg, ”Thick target corrections to K1 production”, KOPIO Technical Note 11, 30 Aug 1998 .
- [4] [http://pubweb.bnl.gov/people/e926/private/KOPIO\\_parameters.html](http://pubweb.bnl.gov/people/e926/private/KOPIO_parameters.html)
- [5] A.Poblaguev, “Progress on beam simulations”, <http://pubweb.bnl.gov/users/e865/www/KOPIO/Collimators/note080701.ps>.
- [6] D. Bryman, ”E926 Preradiator Study: Alternating Horizontal and Vertical Anode Wires and Cathode Strips” , KOPIO Technical Note 14, 25 March 1999 and 14 April 2003 addendum.
- [7] G. S. Atoyan, V. V. Issakov, A. A. Poblaguev, M.E. Zeller, O.V. Karavichev, T.L. Karavicheva, ”Study of Honeycomb Strip Chamber Preradiator for KOPIO”, KOPIO Technical Note 16, 3 Nov 2000.
- [8] F. Marcucci, ” S/N vs signal events for several energy resolutions” KOPIO Technical Note 25, 21 September 2001.
- [9] Appel *et al.*, Nucl. Instr. and Meth. **A479** (2002) 349.

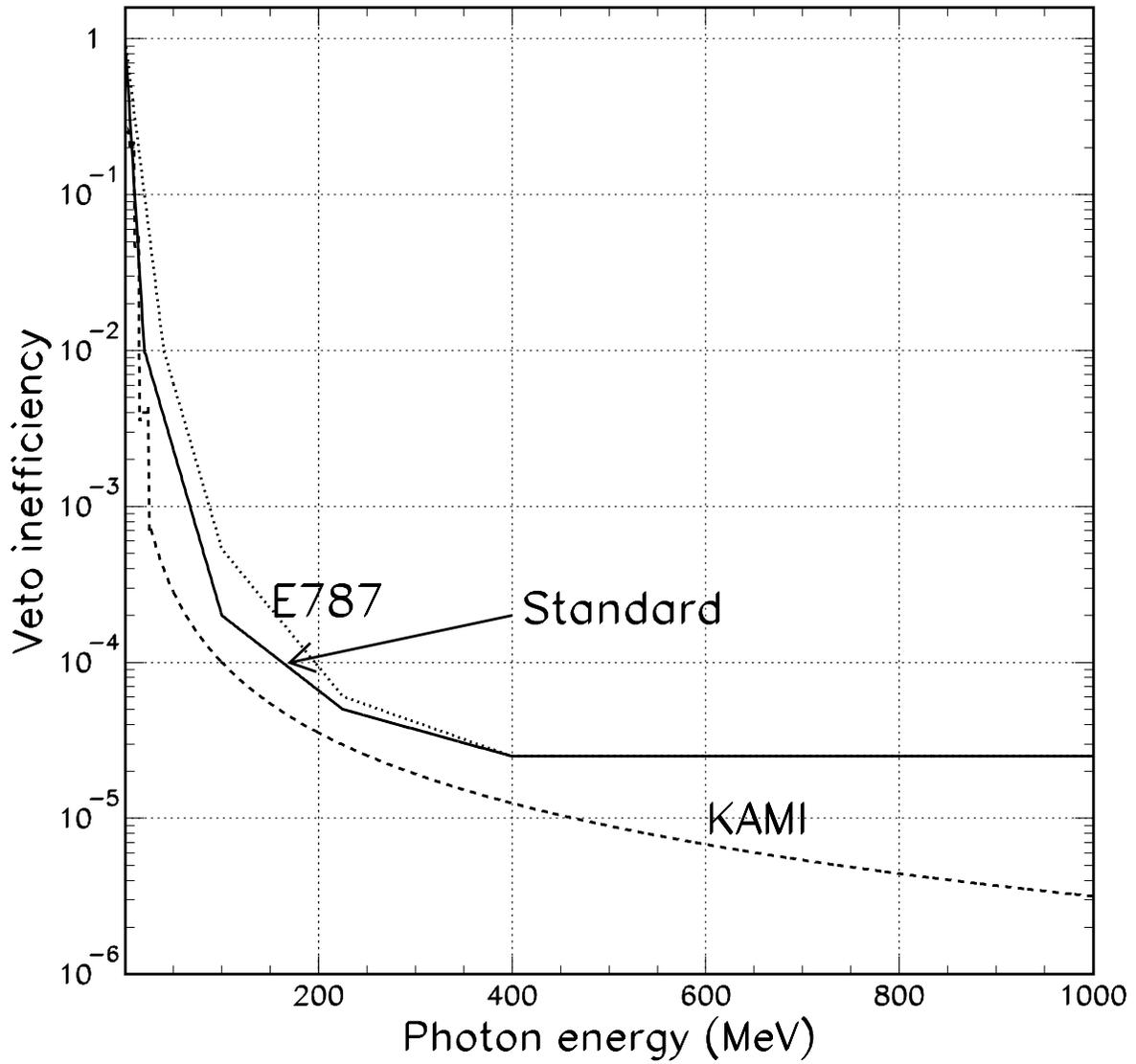


Figure 3: The photon veto inefficiencies as a function of photon energy in MeV/c used in analysis. The standard photon veto inefficiency as well as approximations to the veto inefficiency of E787 and KAMI are shown.

```

      REAL FUNCTION pv(Egam)
      IMPLICIT NONE
* 21 Dec 2000 djaffe
* Akira's photon veto INefficiency function
* modified so that input argument is photon energy in MeV
*
      Real Egam, Effy

      if(Egam.lt.20) then
         Effy =10000*10**(-Egam/10)
      elseif(Egam.lt.100) then
         Effy =100*50**(-(Egam-20)/80)
      elseif(Egam.lt.225) then
         Effy =2*4**(-(Egam-100)/125)
      elseif(Egam.lt.400) then
         Effy =0.5*2**(-(Egam-225)/175)
      else
         Effy =0.25
      endif

* explicitly add Akira's scale factor
pv=Effy*1.e-4
if (pv.gt.1.) print *, ' PV:effy,egam=', effy*1.e-4, egam
END

```

Figure 4: The explicit form of the standard PV function.

```

      Real Function ChgVeto(p,ID)
c      =====
c      determine charged veto 1-effy for input momentum P
c      ID = 1,2,3   = e+,pi+,mu+
c      -1,-2,-3 = e-,pi-,mu-
c      For mu+,mu- assume 1.e-5
c      For e-, pi+ use Inagaki KEK94-125
c      For e+ use 0.1/E(MeV) from A.Konaka
c      For pi- use 3e-4 + 3e-2/max(200.,P) P in MeV from A.Konaka
c      03may01 Scale factor for charge vetos implemented
      IMPLICIT NONE
c      arguments
      Real P
      Integer ID
c      local
      Real em2
      parameter (em2 = 0.510998902 * 0.510998902)
      Real effbar
      Logical FIRST/.TRUE./
c      02may01 improve all charge vetos by Scale (X2)
      Real Scale/2.0/
      Integer i
      Integer NE
      Parameter (NE=3)
      Real Ineff(-NE:NE)
c      Data Ineff/1.e-5, -1., 1.3e-4, -1., -1., 1.6e-5, 1.e-5/
c      mu- , pi-, e- , X , e+, pi+ , mu+
c      17april01 : change ineff(pi+) from 16e-6 to 3e-6 (Akira rightly
c      points out that Inagaki's ineff(pi+) is a limit and
c      results for higher momentum pi+ indicate that 3e-6
c      is achievable
      Data Ineff/1.e-5, -1., 1.3e-4, -1., -1., 0.3e-5, 1.e-5/
c      mu- , pi-, e- , X , e+, pi+ , mu+
      Save Ineff,FIRST

      if (abs(ID).gt.NE .or. ID.eq.0) then
        print *,' CHGVETO: Invalid ID=',ID,' ABORT!!!'
        STOP ' CHGVETO: Invalid ID'
      endif

```

Figure 5: Part 1 of the explicit form of the standard charged veto functions.

```

      if (FIRST) then
c 02may01 improve all charge vetos
      do i = -NE,NE
          if (Ineff(i).gt.0.) Ineff(i) = Ineff(i)/Scale
      Enddo
      print '(1x,a,7(1x,a,g9.3))', 'ChgVeto Ineffys',
&          ' (mu-)=', Ineff(-3),
&          ' (pi-)=', Ineff(-2),
&          ' (e-)=', Ineff(-1),
&          ' (e+)=', Ineff(1),
&          ' (pi+)=', Ineff(2),
&          ' (mu+)=', Ineff(3), 'Scale=', Scale
      FIRST = .false.
    endif

    if (ID.eq.-2) then ! pi-
      effbar = ( 3.e-4 + 3.e-2/max(200.,P) )/Scale ! 02may01
    elseif (ID.eq.1) then ! e+
      effbar = 0.1/sqrt(p*p+em2) / Scale ! = 0.1/E/Scale 02may01
    else
      effbar = Ineff(ID)
    endif

    if (effbar.lt.0.) then
      print *, ' CHGVETO WARNING: 1-effy=', effbar,
&          ' for ID=', ID, ' P=', P
      effbar = 0.
    endif

    ChgVeto = effbar

  END

```

Figure 6: Part 2 of the explicit form of the standard charged veto functions.

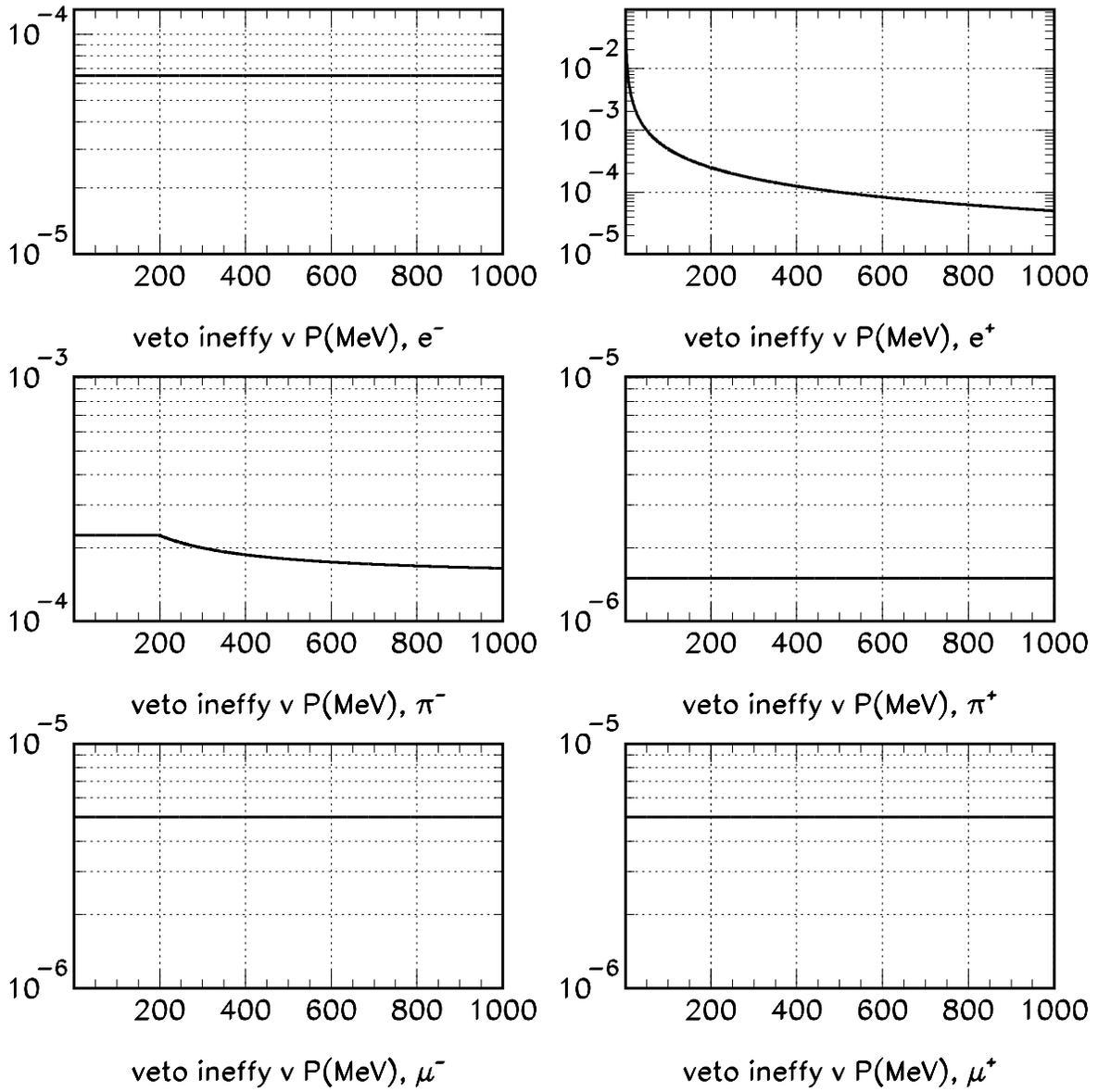


Figure 7: The charged particle veto inefficiencies as a function of momentum in MeV/c used in analysis.

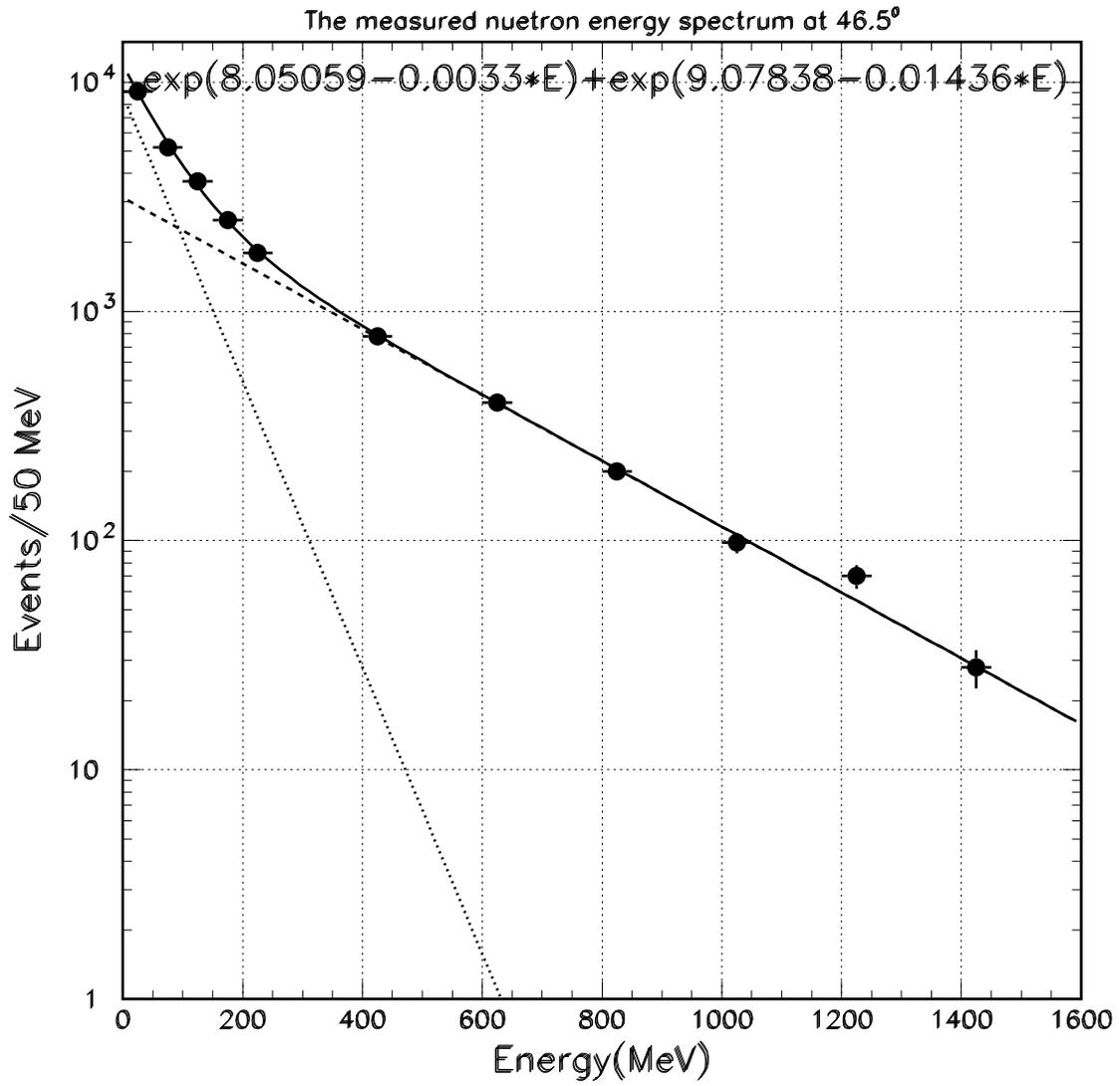


Figure 8: The parametrized neutron energy spectrum.

```

DEBUG 0
* default control file. '*' or '.' in first column is a comment line
* first line must always be DEBUG n where n=0->5 = no debug -> full debug
* (otherwise can't debug this file)
*
* Event source = 0,1 = generate,read from file
EVENT SOURCE 0 ! 0,1 = generate, read from file
* number of events to generate
EVENTS 10000
* Copy from input ntuple to output ntuple? 0=no, otherwise yes
COPY JOB 0
*
* random number seeds and # of events to skip
*     seed1 seed2 nskip1 nskip2
* SEEDS 3185 74 497003 0
SEEDS 8010 1 0 0
* SEEDS 2003 31 319588 0
* SEEDS 1224 3347 0 0
*
***** IMPLEMENT nn->pi0X
***** if N.ne.0, then implement neutrons.
***** if N=2, then use measured neutron spectrum, value of P is irrelevant
***** otherwise use momentum P
*           N   P     x   y   z
NEUTRON BEAM 0 801.  0.000  6.75 1349.0
*
* select KOL beam: 0 = delta function (px=py=0, pz=800 MeV/c)
*                 1 = from KSPEC (realistic)
*                 >1 = from KSPEC with AveKOLperBucket KOL/event
*                 -1 = from KAPINOS 45 degrees (realistic)
*                 <-1 = from KAPINOS with AveKOLperBucket KOL/event
KBEAM -2 0. 0.
* KBEAM 0 300. 150.
*
* COLLIMATION = 0,<>0 = none,use BEAM_COLLIMATION
COLLIMATION 1
*
* EXTENDED_TARGET = 0,<>0 = no, extended target from GEN_VERTEX
EXTENDED_TARGET 1
*

```

Figure 9: Part 1 of the file scripts/control.dat.

```

*
* select K0L decay mode
*   1 pi0,nu,nubar
*   2 pi0pi0
*   3 pi+pi-
*   4 gamma,gamma
*   5 3pi0
*   6 pipipi0
*   7 Ke3
*   8 Kmu3
*   9 Ke3g
*  10 Kmu3g
*  11 pi0,gamma,gamma
*  12 Ke4
*  13 Kmu4
*  14 Kbeta (KL -> K,e,nu)
*  >14 undefined
K0Ldecaymode 1
*
* allow e+ annihilation in vacuum tank?
*   0 = no, 1=true,2=approx
ANNIHILATION 0
*
* select conversion point and PRERADIATOR MODEL
*   0 = Z(rear CAL, AK PR model)
*   1 = Z(middle of PR, AK PR model),
*   2 = Z(middle of PR, M.Zeller PR)
*   3 = Z(rear PR, AK PR model)
CONVERSION POINT 0
*
* if V.ne.0, enable weighting. delZi -> Zfiducial wrt decay region
*   US end = Z(US end of decay region) - delZ1
*   DS end = Z(DS end of decay region) + delZ2
*           V delZ1 delZ2
* USE K WEIGHT 1 0. 0.
* USE K WEIGHT 1 -100. 000.
*** for decays in between z=400 and 800 cm
USE K WEIGHT 1 0. -0.
*
* Force all charged pions that decay to go thru pi0,e,nu?
*   0 = no, 1 = YES
PION BETA DECAY 0
*
* Maximum cos(KL,e) in lab for KE4 decay (No cut if >1., obviously)
MAXIMUM COS(K,e) FOR KE4 2.0
*
* incorporate resolution effects?
*   0 = no, 1 = smear as CAL conversion, 2 = smear as PR conversion
* SMEARING 2
SMEARING 0
*

```

Figure 10: Part 2 of the file scripts/control.dat.

```

* do constrained fits to pi0 candidates?
*      0 = no, otherwise yes
* PIO FIT  1
PIO FIT 0
* which fits to do? 0=no, otherwise yes
*      2g  1g 0g
WHICH FITS?  1  0  0
*
* trigger requirements: i1,i2 = min,max # of hits in PRCAL
*      j1,j2 = min,max # of hits in decay volume veto region
* setting i1,j1 < 0 and i2,j2 > 1000 will accept all events
* format: TRIGGER i1 i2 j1 j2
TRIGGER -1 1000 -1 1000
*
* Skim requirements (ma => MAXimum, mi => MInimum)
* SKIM ACTIVATE 1
SKIM ACTIVATE 0
*      0 = no skim, otherwise do skim
SKIM PHOTON  300. 300. 0. 0.
*      xma yma xmi ymi : xma,yma at ~10 RL into CAL. xmi,ymi at front of CAL
SKIM KAON  0. 1500. 0. 2000.
*      Emi Ema Zmi Zma : FITTED Energy, Z-position of KL
SKIM PION 100. 170. 0. 250. 0.
*      Mmi Mma E*mi E*ma Oddmi : Fitted Mgg, E*(pi0), OddMin
*      OddMin = E*(pi0) - |E*g1 - E*g2|
SKIM KP2  0
*      n = 0,1,2 = all decays, Kp2 Odd only, Kp2 Even Only accepted
*
*
*
** 15aug01 output ntuple type = 0,1,2,3 = none,jaffe-style,konaka-style,both styles
** jaffe-style and konaka-style ntuples are put in different output files
NTUPLE TYPE 1
*
* input ntuple files and time offsets in ns
*      filename      deltaT
* HFIN  /u1/djaffe/kopio/fastmc/ntuples/kp2.2902.21.no.smear.not.fit.rzn 0.
*
* output histogram file name (max of 132 characters)
* if multiple lines with HFNAME, use them in order
* HFNAME /u1/djaffe/kopio/fastmc/ntuples/kp2.2902.21.fit.p0.rzn.entskim
HFNAME /u1/djaffe/kopio/fastmc/ntuples/kpnn.kapinos2.extended.tgt.coll.rzn
*
*      1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
* end of control file

```

Figure 11: Part 3 of the file `scripts/control.dat`.

```

* comment lines start with '*'
DEBUG 0          ! 0=none, 5=max
EVENT SOURCE 0   ! = 0,1 = generate,read from file
EVENTS 200000    ! number of events to generate
SEEDS 21064 21 0 0 ! seed1 seed2 nskip1 nskip2
KBEAM -2. 0. 0.   ! = 0,1,-1,-2 = delta fcn pz=800MeV, from KSPEC, from KAPINOS, fro
USE K WEIGHT 1 0. 0. ! V delZ1 delZ1 : if V.ne.0, enable weighting. delZi -> Zfiducia
KOLdecaymode 2    ! 1-13 = pnn, 2pi0, 2pi, gg, 3pi0, 2pipi0, Ke3, Kmu3, Ke3g, Kmu3g,
ANNIHILATION 0    ! allow e+ annihilation in vacuum tank? 0 = no, 1=true,2=approx
CONVERSION POINT 2 !=0,1,2 = Z(rear CAL), Z(middle of PR), Z(M.Zeller PR) of gamma con
MAXIMUM COS(K,e) FOR KE4 1.1 ! max cos_lab(KL,e) allowed for Ke4 decays
SMEARING 2       !incorporate resolution effects? 0=no, 1/2=smear as CAL/PR conv.
PIO FIT 1        !do constrained fit to pi0 cand? 0=no, otherwise yes
WHICH FITS? 1 0 0 !11jun03 2g fit enable, 1g, 0g fits disabled
FORCE YBEAM HEIGHT IN FIT -10. ! 14nov02 force beam size in pi0 fit constraint
SKIM ACTIVATE 1  ! 0 = no skim, otherwise do skim
TRIGGER -1 10 -1 10 ! this accepts all events
* HFIN filename 0. ! input ntuple file(s) and time offset(s) in ns
* output hist file name (max of 80 char)
* HFNAME filename
* 15apr04 add two elements
COLLIMATION 1    ! = 0,<>0 = none,use BEAM_COLLIMATION
EXTENDED_TARGET 1 ! = 0,<>0 = no, extended target from GEN_VERTEX
HFNAME ../localntuples/kp2.21064.21.rzn.entskim

```

Figure 12: The file `condir/control.kp2.21064.21` created by `scripts/run.fastmc.script` for  $K_L^0 \rightarrow \pi^0 \pi^0$  production.

```

* detector.21.dat = fastMC detector description
* 16apr04 copy of detector.13 with larger US beampipe
*     AND with decay volume moved from (950,1350) to (1015,1415) cm
*     AND larger DS beam hole (as in detector.18)
* For aspect ratio#1 of Jaap's TN049 note
* '*' or '.' in first column is a comment line
*
* cartoon of detector as of 12 dec 2000 useful for
* definition of lengths and widths given below.
* All volumes are rectangular parallelpipeds.
*
*
* -----
*          |                |PR|  | Catcher  |
*          | DecayVolume    | |CAL| Volume  |
* x --Zbeam-----=          = = =          > catcher
*          |                | |  |  |
*          |                | |  |  |
* -----
*
* Units of fastMC are cm, MeV, ns
* RH coordinate system with z = KOL beam direction, y = up
* origin is the production point of KOL beam: (0.,0.,0.)
*
* half width and height of beam pipe and distance from
* KOL production point to US end of decay volume.
* This is also the size of the DS beam hole through
* the preradiator and calorimeter.
* Half width & height and length of decay volume,
* preradiator, calorimeter and catcher volume
*(note that catcher volume isn't catcher and that
* 'length' of catcher is irrelevant as MC only checks
* impact position at front of catcher)
* Available materials are 1 = vacuum, 2 = non-vacuum
* 27feb01: adjust dimension of beam pipes, catcher volume and detector
* 30apr01: adjust dimensions of pipe, decay volume, prerad...
* 1oct02: KL beam divergence is now adjustable parameter
* 10oct02: clearance at front of PR is now adjustable

```

Figure 13: Part 1 of the file geometry/detector.21.dat.

```

* KOPIO parameters from TDR 30apr01 ( http://pubweb.bnl.gov/users/e926/www/private/
* BEAM PIPE from TDR Use
* beginning of decay volume 950cm 950cm
* beam width at DS edge of US veto wall 95cm 100cm
* beam height 4.75cm 6cm
* DECAY VOLUME from
* length of decay volume 400cm 400cm
* PR from
* preradiator depth 100cm 100cm
* active tracking area of PR rear face 400cm 400cm (x & y)
* CAL from
* calorimeter depth 80cm 80cm
* CAL total width 517 500
* CAL total height 517 500
* DS HOLE from
* length must match PR + CAL length
* eyepiece maximum width 170cm 170cm
* eyepiece maximum height 21.3cm 23.3cm
* CATCHER VOLUME from
* position of catcher face 2650cm 2650cm (=950+400+100+80+1120)
* catcher module horizontal size 400cm 400cm
* vertical size arbitrary (if > vertical size of catcher)
* CATCHER DETECTOR from
* depth in Z is arbitrary (>0)
* catcher module horizontal size 400cm 400cm
* catcher module vertical size 20cm 20cm
* x y z material
*BEAM PIPE 50. 3. 950. 1
* BEAM PIPE 35. 4. 950. 1
* BEAM PIPE 65. 5. 950. 1
BEAM PIPE 65. 5. 1015. 1
DECAY VOLUME 200. 200. 400. 1
PRERADIATOR 200. 200. 100. 2
CALORIMETER 250. 250. 80. 2
*DOWNSTREAM HOLE 85. 10.65 180. 1
*DOWNSTREAM HOLE 64. 11. 180. 1
* 16apr04 larger DS hole to be consistent with GEANT MC v07_4
DOWNSTREAM HOLE 110.25 11.55 180. 1
CATCHER VOLUME 200. 50. 1120. 1
CATCHER DETECTOR 200. 10. 0.01 1

```

Figure 14: Part 2 of the file geometry/detector.21.dat.

```

*
* specify beam divergence in RADIANS. Note that this is the full opening angle in radi
* of the beam not the half-angle. (slight inconsistency with detector element
* specifications which use half-size in x and y).
*           x       y
BEAM DIVERGENCE 0.100 0.005
* BEAM CLEARANCE 25.6 5.36
* 16apr04 MAKE CLEARANCE NEGATIVE TO FORCE USAGE OF 'BEAM PIPE' and 'DOWNSTREAM HOLE'
BEAM CLEARANCE -25.6 -5.36
*
* add 'clearance' around beam at front of PR in CM
* clearance = (aperture size - beam size)/2
* if clearance < 0 or the 'BEAM CLEARANCE' line is missing, then
* the dimensions are taken as specified above.
*
*
** dimensions below used from 27feb-30apr01
**           x       y       z       material
*BEAM PIPE           60.   3. 1000.   1
*DECAY VOLUME       200. 200.  400.   1
*PRERADIATOR       200. 200.   61.   2
*CALORIMETER       250. 250.   60.   2
*DOWNSTREAM HOLE   85.  4.5 121.   1
*CATCHER VOLUME   130.  10. 1100.   1
*CATCHER DETECTOR 130.   7.   0.1   1
*
*           Z       A  density(g/cm^3)  thickness(cm)
*** VACUUM TANK  6. 12.011 2.265           1.3
VACUUM TANK  6. 12.011 2.265           3.528
*
* preradiator/calorimeter resolutions (used in smear.f)
*           C_E : sigma/E = C_E/sqrt(E) , E in GeV
ENERGY RESOLUTION 0.027
*
*           S_Txz  S_Tyz  f_angle  r_angle  rho_angle
* ANGULAR RESOLUTION 8.2e-3 8.2e-3 1.00    3.6    0.
ANGULAR RESOLUTION 8.2e-3 8.2e-3 0.96    3.6    0.
*           [rad GeV^0.7]
*           x       y  in [cm sqrt(GeV)]
POSITION RESOLUTION 0.45 0.45
*           ns
TIME RESOLUTION 0.2
*
* ----- end of detector.dat

```

```

c information about neutral or charged 'tracks' in current event.
c 4momentum = px,py,pz,E
c position = x,y,z : z = KOL beam direction,
c           y = up,
c           x = left facing downstream
c           x,y,z = 0,0,0 at KOL production target
c units are MeV, cm, ns
c Integer Mtrack
c if Mtrack changes, then ntuple specifications must be modified below and in reconinfo.inc
c   Parameter (Mtrack = 600) ! max # of tracks/event
c   Integer Ntrack ! current # of tracks
c   Integer Ipart ! particle type
c   Real Charge
c   Integer Parent ! parent of track. = 0 if no parent
c   Real PinKCMS ! 4momentum in KOL center-of-momentum
c   Real Vbirth ! point in space of track birth
c   Real Vdeath ! idem          death
c   Real Pbirth ! lab 4momentum of track at birth
c   Real Pdeath ! idem          death
c   Real Tbirth ! time of track birth (KOL birth = 0)
c   Real Tdeath ! idem          death
c   Integer DecayMode ! decay mode of track (0 if none)
c   Integer Fate ! fate of track = (1-10) allowed, 1,2=decay,hit used
c   Integer LastVol ! volume where track died (1-Mvolume) allowed
c   Integer LastPL ! PLANE # (1-6) where track died
c   Real Weight ! associate with decay of particle
c copies of 2d arrays because HBOOK apparently can not handle
c 2d arrays in CWN despite statements to the contrary in the manual
c   Real PxBirth,PyBirth,PzBirth,Ebirth
c   Real PxDeath,PyDeath,PzDeath,EDeath
c   Real xBirth,yBirth,zBirth
c   Real xDeath,yDeath,zDeath
c   Real PxinKCMS,PyinKCMS,PzinKCMS,EinKCMS
c 7aug01 Smeared(itk) = 0 => track itk was not smeared
c           1 =>          smeared as if converted in CAL
c           2 =>          smeared as if converted in PR
c   Integer Smeared
c ntuple variables
c   COMMON/TrackInfo/
c   & Ntrack,
c   & Ipart(Mtrack), Charge(Mtrack),
c   & Parent(Mtrack), Tbirth(Mtrack),
c   & Tdeath(Mtrack), DecayMode(Mtrack),
c   & Fate(Mtrack), LastVol(Mtrack),
c   & LastPL(Mtrack), PxinKCMS(Mtrack),
c   & PyinKCMS(Mtrack), PzinKCMS(Mtrack),
c   & EinKCMS(Mtrack), PxBirth(Mtrack),
c   & PyBirth(Mtrack), PzBirth(Mtrack),
c   & EBirth(Mtrack), PxDeath(Mtrack),
c   & PyDeath(Mtrack), PzDeath(Mtrack),
c   & EDeath(Mtrack), xBirth(Mtrack),
c   & yBirth(Mtrack), zBirth(Mtrack),
c   & xDeath(Mtrack), yDeath(Mtrack),
c   & zDeath(Mtrack), Weight(Mtrack),
c   & Smeared(Mtrack) ! add 7aug01

```

Figure 16: Per-particle info stored in trackinfo.inc.

```

c --- reconinfo.inc
c if you change Mrecon, you must change CHrecon (for ntuple)
  Integer Mrecon
  Parameter (Mrecon = 50) ! 23aug01 increase from 15 to 50
  Integer Nrecon
  Real RecMass      ! M(gamma1,gamma2)
  Real RecE1inCMS ! E(gamma 1) in KOL CMS
  Real RecE2inCMS ! E(gamma 2) idem
  Real RecEpinCMS ! E(pi0) idem
  Real DOCA        ! distance of closest approach between gamma1,2 trajectories
  Real RecT1       ! T(gamma 1) at DOCA
  Real RecT2       ! T(gamma 2) idem
  Logical GoodPair! = true if gamma1,gamma2 are daughter of same pi0
  Real RecPxK,RecPyK,RecPzK,RecEK      ! reconstructed KOL 4momentum
  Real RecXK,RecYK,RecZK ! reconstructed KOL position
  Real RecMM,RecEmiss ! reconstructed missing mass (signed), missing energy
  Real RecE1,RecE2 ! E(gamma1),E(gamma2) in lab
c constrained fit information
c error code from constrained fits:
c = 0 : both fits ok
c > 0 : at least one fit failed
c < 0 : fits not attempted
  Integer RecIerr ! error code from constrained fit
  Real RecChi2NoM ! chi2 from 1st constrained fit
  Real RecChi2 ! chi2 from 2d constrained fit
  Integer RecIter ! # of iterations in 2d constrained fit
  Real RecProbNoM ! Prob(chi2,ndf) from 1st constrained fit
  Real RecProbM !prob(chi2,ndf) from 2d constrained fit (with Mpi0 constraint)
c pointer to /TrackInfo/ track #
  Integer RecItk1
  Integer RecItk2
c position of reconstructed photons AFTER constrained fit
  Real RecX1,RecY1,RecZ1
  Real RecX2,RecY2,RecZ2
c 1feb01: distance between e+ and e- of conversion in PR at
c last measurement layer (only for ConvFlag=2)
  Real RecXdif1,RecYdif1
  Real RecXdif2,RecYdif2
c 2mar01: add fitted momenta of photons
  Real RecPx1,RecPy1,RecPz1
  Real RecPx2,RecPy2,RecPz2

```

Figure 17: The first part of the per-candidate information.

```

c 23aug01 Add 'type' of fit: 0 = 2g converted in PR
c                               1 = g1 converted in PR, g2 converted in CAL
c                               2 = both g converted in CAL 2jan03
Integer FitType

Real RecPK      ! reconstructed KOL 4momentum
Real RecVK      ! reconstructed KOL position
Logical RecSkim ! = true, if candidate accepted by skim cuts
COMMON/ReconInfo2/RecPK(4,Mrecon),RecVK(3,Mrecon),RecSkim(Mrecon)

c for column-wise ntuple
Integer LCHrecon
Parameter (LCHrecon=
& 13+35+38+41+19+30+29+42+31+28+43+40+36+50+41+43+34+34+45+45+22
& )
Character CHrecon*(LCHrecon)
Parameter (CHrecon=
& 'Nrecon[0,50], '//
& 'RecMass(Nrecon),RecE1inCMS(Nrecon), '//
& 'RecE2inCMS(Nrecon),RecEpinCMS(Nrecon), '//
& 'DOCA(Nrecon),RecT1(Nrecon),RecT2(Nrecon), '//
& 'GoodPair(Nrecon):L, '//
& 'RecPxK(Nrecon),RecPyK(Nrecon), '//
& 'RecPzK(Nrecon),RecEK(Nrecon), '//
& 'RecXK(Nrecon),RecYK(Nrecon),RecZK(Nrecon), '//
& 'RecMM(Nrecon),RecEmiss(Nrecon), '//
& 'RecE1(Nrecon),RecE2(Nrecon), '//
& 'RecIerr(Nrecon)[-1,7]:I,RecChi2NoM(Nrecon), '//
& 'RecChi2(Nrecon),RecIter(Nrecon)[0,20]:I, '//
& 'RecProbNoM(Nrecon),RecProbM(Nrecon), '//
& 'RecItk1(Nrecon)[0,600]:I,RecItk2(Nrecon)[0,600]:I, '//! increase if Mtrack
& 'RecX1(Nrecon),RecY1(Nrecon),RecZ1(Nrecon), '//
& 'RecX2(Nrecon),RecY2(Nrecon),RecZ2(Nrecon), '//
& 'RecXdif1(Nrecon),RecYdif1(Nrecon), '//
& 'RecXdif2(Nrecon),RecYdif2(Nrecon), '//
& 'RecPx1(Nrecon),RecPy1(Nrecon),RecPz1(Nrecon), '//
& 'RecPx2(Nrecon),RecPy2(Nrecon),RecPz2(Nrecon), '//
& 'FitType(Nrecon)[0,2]:I'
& )!           1           2           3           4           5
c           12345678901234567890123456789012345678901234567890123456789
c -----

```

Figure 18: The second part of the per-candidate information.

```
c 26aug04 dv Add more variables
```

```
Real RecP1inCMS, RecP2inCMS ! photons momenta in CMS
Real RecX4Kf1                ! Kaon (x,y,z,t) after 1st fit

Logical RecPropOK           ! .true. if valid propagated errors are written
                             ! to variables below
Real RecCovX4Kf1           ! Covariance matrix (CM) of above variable
Real RecCovX4K             ! As above but after 2nd fit

Real RecCov1                ! CM of (E*,E1*-E2*,Emiss,Hpar)
                             ! Hpar is  $\text{SQRT}(E^2 - (E1 - E2)^2)$ 
Real RecSmass               ! Sigma on reconstructed mass (after 1st fit)

Real RecTK                  ! reconstructed t of Kaon after 2d fit

Logical GoodARes1          ! .true. if narrower gaussian was used to play angle
Logical GoodARes2          ! the same for 2nd photon
```

```
c 4oct04 djaffe Add RecValidKL
```

```
Logical RecValidKL        ! .true. if  $\beta(KL) < 1$ 
```

Figure 19: Third part of the per-candidate information.