Présentation 3 γ

École IN2P3 d'informatique 2016

« Parallélisme sur matériel hétérogène »

Vincent Lafage

S2I, Institut de Physique Nucléaire
Université d'Orsay

23 Mai 2016

- Lexique, rapide, de physique des particules
- Implémentation
- Optimisation : cache-miss, profiling
- vers une version parallèle

« New physics with three photon events at LEP »,
M. Baillargeon, F. Boudjema, E. Chopin, V. Lafage,

Z.Phys. C71 (1996) 431–442, e-Print: hep-ph/9506396

Éléments de lexique (i)

« Le lexique est surprenant... mais la syntaxe est correcte »

- un **collisionneur**
- ... prépare des **faisceaux**
- ... qui correspondent à un **état initial** bien défini : **énergie**, **polarisation**
- ... de deux types de **particule**

Dans notre cas

- un faisceau d'**électrons** e^-
- un faisceau de **positrons** e^+
- **tête-bêche**
- **d'énergie** E identique

Éléments de lexique (ii)

- les physiciens ont mis en place un système de **détecteurs**
- ... pour enregistrer des **événements**
- ... parmi lesquels ils peuvent identifier des **états finals**
- ... dans des **configurations** géométrique & d'énergie donnée

Dans notre cas

- tous les détecteurs installés sur le LEP était capable d'étudier ce processus, surtout L3
- l'état final **3 photons** γ

Éléments de lexique : détecteurs

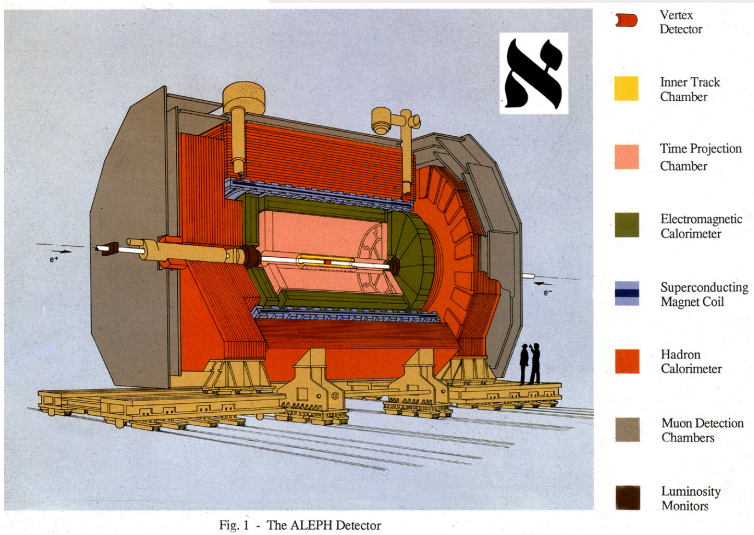
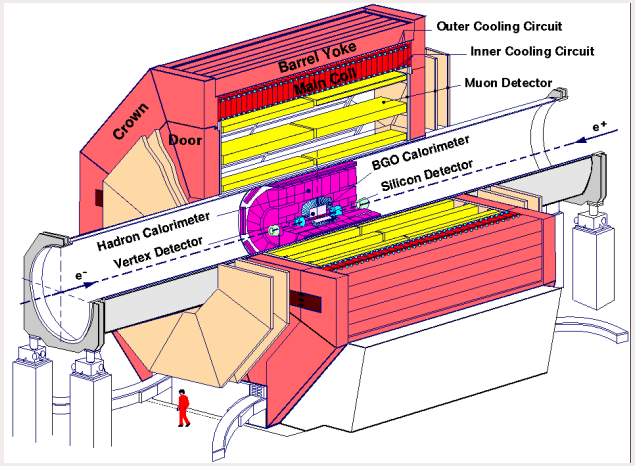


Fig. 1 - The ALEPH Detector



Comme il y a des contraintes sur le détecteurs, (géométrie, passage du tube faisceau, passage des cables…), il y a des zones aveugles. Leur modélisation est un travail de précision (*cf* `Geant`). Ici, modélisation grossière par des **coupures** :

- en dessous d'un certain angle par rapport au faisceau, on ne verra pas la particule (perdue dans le tube)
- en dessous d'un certain angle entre deux particules, on ne les distinguera pas : coupure de **séparation**
- en dessous d'une certaine énergie, on n'identifiera pas la particule

Éléments de lexique : comptage

Combien d'événement espérons nous compter ? Comment prévoir le collisionneur et les détecteurs pour engranger une statistique suffisante ? Y en aura-t-il assez pour discriminer ce qui est standard, et ce qui est de la nouvelle Physique ?

- Plus longtemps on a du faisceau, plus on doit voir d'événements: ***taux de production*** N

$$\begin{aligned} \mathcal{N} &\propto T & (1)\\ &= N \times T & (2) \end{aligned}$$

- Si on pousse le faisceau (plus d'intensité, plus de focalisation,...) on verra plus d'événements : on résume tout ça dans $\mathcal{L}$ la ***Luminosité*** instantanée.

$$\begin{aligned} N &\propto \mathcal{L} & (3)\\ &= \mathcal{L} \times \sigma & (4) \end{aligned}$$

le coefficient de proportionnalité σ ne dépend plus du collisionneur, mais *seulement de la physique*, au moins celle qui tombe dans les coupure... C'est la ***section efficace***

« Petite section efficace... grosse luminosité », proverbe Shadock

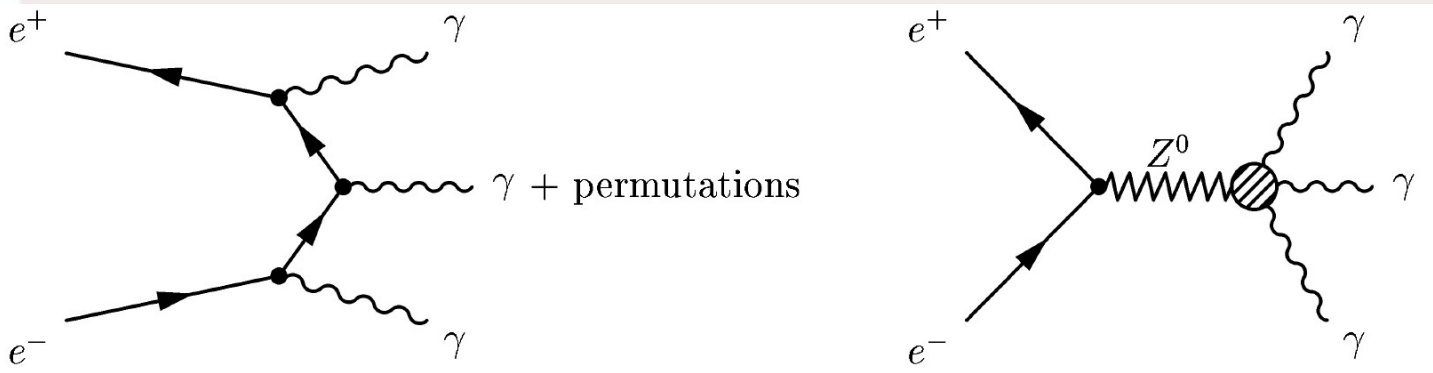
Éléments de lexique : section efficace

Comment la calculer ?

- d'autant plus grande qu'ils y a plus de configurations possibles pour les particules sortantes : somme sur l'espace des configurations, appelé ***espace des phases***

$$\sigma = \int |\mathcal{M}|^2 \times \delta^4(P - p_1 - \cdots - p_n) \prod_{i=1}^{n} \mathrm{d}^4 p_i \, \delta(p_i^2 - m_i^2) \, \theta(p_i^0) \quad (5)$$

- ce qu'on somme (la section efficace différentielle) dépend de ***l'élément de matrice de transition*** $|\mathcal{M}|^2$ calculé à partir des ***graphes de*** **Feynman**



Polarisation

Ici, chacunes des 5 particules (électron, positron, photons) a deux degrés de liberté de polarisation, ou ***d'hélicité***, d'où $2^5 = 32$ configurations d'hélicité :

$$\begin{aligned}
\mathcal{A}^{QED}_{\pm,\pm;\lambda_1,\lambda_2,\lambda_3} &= 0 \\
\mathcal{A}^{QED}_{+,-;\pm,\pm,\pm} &= 0 \\
\mathcal{A}^{QED}_{+,-;-,-,+} &= -\frac{S(p_1,p_2)S(p_1,k_3)^2}{S(p_1,k_1)S(p_1,k_2)S(p_2,k_1)S(p_2,k_2)} \\
\mathcal{A}^{QED}_{+,-;+,+,-} &= \left(\frac{S(p_1,p_2)S(p_2,k_3)^2}{S(p_1,k_1)S(p_1,k_2)S(p_2,k_1)S(p_2,k_2)}\right)^*
\end{aligned}$$

avec

$$S(p,q) = \frac{(p_1+ip_2)}{\sqrt{(p_0+p_3)}}\sqrt{(q_0+q_3)} - \frac{(q_1+iq_2)}{\sqrt{(q_0+q_3)}}\sqrt{(p_0+p_3)} \quad ;$$

$$p = (p_0, p_1, p_2, p_3).$$

Intégration numérique

$\Rightarrow$ méthode de ***Monte-Carlo***

Ici, la plus simple (pas de raffinement adaptatif) :

- générer un événement avec un certain poids,
  - générateur de nombres pseudo-aléatoires
  - méthode `RAMBO`
    « A new Monte Carlo treatment of multiparticle phase space at high energies »,
    R. Kleiss, W. J. Stirling, S. D. Ellis, Comput. Phys. Commun. 40, 359 (1986).
- voir s'il passe les coupures,
- calculer l'élément de matrice,
- pondérer,
- accumuler...

...et recommencer

Mais il y a deux types d'itérations :

1. celles qui dépendent de la précédente...
2. ... et les autres $\Rightarrow$ Monte-Carlo est ***éhontément parallélisable***
   (*embarassingly parallel*)

Monte-Carlo ?

évaluons une intégrale $$I = \int_{\mathcal{V}} \mathrm{d}x\, f(x)$$

méthode numérique, dite « *des trapèzes* »

$$I_N = \frac{1}{N}\left(\frac{f(x_1)+f(x_N)}{2} + \sum_{i=2}^{N-1} f(x_i)\right) \text{ avec } x_i = \frac{i-1}{N-1}$$

dont la convergence est assez rapide pour 1 dimension.

Considérons désormais des points choisis suivant une distribution $\rho(x)$ à support dans $\mathcal{V}$ ($\int_{\mathcal{V}} \mathrm{d}x\, \rho(x) = 1$).

estimateur à N points de l'intégrale : $$I_N = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{\rho(x_i)}$$

C'est un estimateur ***fidèle*** car son espérance correspond à la valeur de l'intégrale :

$$E(I_N) = \frac{1}{N}\sum_{i=1}^{N} E\frac{f(x_i)}{\rho(x_i)} = \int_{\mathcal{V}} \mathrm{d}x\, \rho(x)\frac{f(x)}{\rho(x)} = I$$

Voyons si I_N est un estimateur ***précis*** en calculant sa variance

$$V(I_N) = \frac{1}{N^2}\sum_{i=1}^{N} V\left(\frac{f(x_i)}{\rho(x_i)}\right) = \frac{1}{N} V\left(\frac{f(x)}{\rho(x)}\right)$$

le code

le code

- les versions de départ

  `hg clone https://[user]@bitbucket.org/bixente/3photons`

  - Fortran 77
  - Fortran 90
  - C++
  - Ada
  - Go

  - Fortran 77 + OpenMP

- la version préparée pour cette école (C99)

```
hg clone https://[user]@bitbucket.org/hyhg/opencl-mpi ./OpenCL-MPI
cd OpenCL-MPI/C99
make
cd MC
./mc
```

…

```
 -------------------------------------------
 Cross-section          (pb) : 11.3022
 Standart Error         (pb) : 0.00885354
 Relative Error              : 0.000783348
 -------------------------------------------
 Elapsed time                : 2.81521
 Elapsed time per event      : 2.81521e-06
 -------------------------------------------
```

MC/input.dat

```
1000000                 'Nombre d''evenements'
91.187e0                'energie dans le centre de masse (GeV)'
0.9e0                   'coupure sur cosinus(photon,faisceau)'
0.9396e0                'coupure sur cosinus(photon,photon)'
4.559e0                 'coupure sur l''energie (GeV)'
0.e0                    'coupure sur sinus(normale,faisceau)'
7.297353079644818e-3    'constante de structure fine'
7.8125e-3               'constante de structure fine au pic du Z'
0.38937966e9            'facteur de conversion (GeV-2 --> pb)'
91.187e0                'masse du Z (GeV)'
2.490e0                 'largeur du Z (GeV)'
0.2319e0                'sinus carre theta Weinberg'
0.03367e0               'taux de branchement Z--->e+e-'
1.0e0                   'Beta +'
1.0e0                   'Beta -'
200                     'Nombre de bins'
.false.                 'Impression des resultats ?'
.false.                 'HBookage des resultats ?'
```

MC/results.dat

```
1   1   1.4140642e+00   3.2107856e-03   2.2706080e-03
1   2   9.5623540e+00   7.3869408e-03   7.7250234e-04
1   3   1.4680267e+01   1.2684984e-02   8.6408402e-04
1   4   0.0000000e+00   0.0000000e+00   8.1662919e-04
1   5  -3.2571012e-04   1.2227974e-03   3.7542505e+00

2   1   1.4140642e+00   3.2107856e-03   2.2706080e-03
2   2   7.1543913e+00   5.5267840e-03   7.7250234e-04
2   3   1.0983527e+01   9.4906902e-03   8.6408402e-04
2   4  -0.0000000e+00   0.0000000e+00   8.1662919e-04
2   5   2.8173136e-04   1.0576901e-03   3.7542505e+00

    1   7.0703211e-01   1.1351841e-03   1.6055623e-03
    2   4.1791863e+00   2.3064073e-03   5.5187952e-04
    3   6.4159485e+00   3.9606030e-03   6.1730593e-04
    4   0.0000000e+00   0.0000000e+00            -nan
    5  -1.0994691e-05   4.0419223e-04   3.6762492e+01
```

Rapide ! Efficace ?

```
time valgrind --tool=cachegrind --branch-sim=yes mc
…
==32423==
==32423== I   refs:      21,866,876,065
==32423== I1  misses:           90,912
==32423== LLi misses:            2,966
==32423== I1  miss rate:          0.00%                                    => 4,2.10^-6
==32423== LLi miss rate:          0.00%
==32423==
==32423== D   refs:      10,907,321,258  (7,120,262,982 rd   + 3,787,058,276 wr)
==32423== D1  misses:           17,778  (       15,391 rd   +         2,387 wr)
==32423== LLd misses:            9,880  (        8,407 rd   +         1,473 wr)
==32423== D1  miss rate:           0.0% (          0.0%     +           0.0%  )    => 1,6.10^-6
==32423== LLd miss rate:           0.0% (          0.0%     +           0.0%  )
==32423==
==32423== LL refs:              108,690  (      106,303 rd   +         2,387 wr)
==32423== LL misses:             12,846  (       11,373 rd   +         1,473 wr)
==32423== LL miss rate:            0.0% (          0.0%     +           0.0%  )
==32423==
==32423== Branches:         920,763,441  (  853,502,037 cond +    67,261,404 ind)
==32423== Mispredicts:       48,791,446  (   42,788,533 cond +     6,002,913 ind)
==32423== Mispred rate:             5.2% (          5.0%     +           8.9%   )
```

-O3 -march=native -funroll-loops

```
 Elapsed time per event         : 1.87726e-06       => -1/3
…
==20432== I   refs:      8,807,755,982
==20432== I1  misses:            16,437
==20432== LLi misses:             2,828
==20432== I1  miss rate:           0.00%
==20432== LLi miss rate:           0.00%
==20432==
==20432== D   refs:      3,307,797,849  (1,911,804,779 rd   + 1,395,993,070 wr)
==20432== D1  misses:            17,650  (       15,274 rd   +         2,376 wr)
==20432== LLd misses:             9,847  (        8,376 rd   +         1,471 wr)
==20432== D1  miss rate:            0.0% (          0.0%     +           0.0%  )
==20432== LLd miss rate:            0.0% (          0.0%     +           0.0%  )
==20432==
==20432== LL refs:               34,087  (       31,711 rd   +         2,376 wr)
==20432== LL misses:             12,675  (       11,204 rd   +         1,471 wr)
==20432== LL miss rate:             0.0% (          0.0%     +           0.0%  )
==20432==
==20432== Branches:         701,657,379  (  642,396,799 cond +    59,260,580 ind)
==20432== Mispredicts:       30,876,538  (   30,873,904 cond +         2,634 ind)
==20432== Mispred rate:             4.4% (          4.8%     +           0.0%   )
```

Profiling

```
vi make.inc

…                                                    …
Common flags                                         Common flags
DFLAGS= -g -m64 -fPIC                    ⇒           DFLAGS= -g -m64 -fPIC -pg
…                                                    …
…
 ------------------------------------------
 Cross-section          (pb) : 11.3022
 Standart Error         (pb) : 0.00885354
 Relative Error              : 0.000783348
 ------------------------------------------
 Elapsed time                : 5.04738
 Elapsed time per event      : 5.04738e-06
 ------------------------------------------
…
 Elapsed time per event      : 4.43646e-06          => avec optimisation
…

gprof ./mc|tee GPROF|less
```

Flat profile

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self us/call	total us/call	name
17.75	0.33	0.33	49579320	0.01	0.01	gsl_complex_mul
14.79	0.61	0.28	708276	0.39	1.39	computeME2
9.14	0.78	0.17	1000000	0.17	0.39	computeSpinorProducts
8.07	0.93	0.15	46914488	0.00	0.00	gsl_complex_mul_real
6.45	1.05	0.12	4249656	0.03	0.03	gsl_complex_div
5.92	1.16	0.11	1000000	0.11	0.24	generateRambo
5.38	1.26	0.10	25666208	0.00	0.00	gsl_complex_conjugate
4.57	1.34	0.09	12000000	0.01	0.01	ocl_rng_rand48_nextState
3.76	1.41	0.07	1000000	0.07	0.07	selectEvent
3.23	1.47	0.06	1000000	0.06	0.07	computeScalarProducts
3.23	1.53	0.06	2124828	0.03	0.05	anomalousAmplitudePMM
2.69	1.58	0.05	20000000	0.00	0.00	gsl_complex_negative
2.69	1.63	0.05	708276	0.07	0.07	updateStatistics
2.15	1.67	0.04	12000000	0.00	0.01	ocl_rng_rand48_getDouble
1.61	1.70	0.03	10000000	0.00	0.00	gsl_complex_sub
1.61	1.73	0.03	2124828	0.01	0.08	QEDAmplitudePMM
1.61	1.76	0.03	1000000	0.03	0.03	sortPhotons
1.08	1.78	0.02	26998624	0.00	0.00	gsl_complex_abs2
1.08	1.80	0.02	2124828	0.01	0.08	QEDAmplitudePPM
1.08	1.82	0.02	708276	0.03	0.08	anomalousAmplitudePPP
1.08	1.84	0.02				main
0.54	1.85	0.01	4249656	0.00	0.00	gsl_complex_abs
0.54	1.86	0.01	1000000	0.01	0.01	resetME2
0.00	1.86	0.00	2833104	0.00	0.00	gsl_complex_add
0.00	1.86	0.00	2124828	0.00	0.02	anomalousAmplitudePPM
0.00	1.86	0.00	708276	0.00	0.05	anomalousAmplitudeMMM

INSTITUT DE PHYSIQUE NUCLÉAIRE ORSAY

-O3 -march=native -funroll-loops

% time	cumulative seconds	self seconds	calls	self ns/call	total ns/call	name
30.78	0.44	0.44	46914488	9.38	9.38	gsl_complex_mul_real
11.19	0.60	0.16	20000000	8.00	8.00	gsl_complex_negative
10.49	0.75	0.15	10000000	15.00	15.00	gsl_complex_sub
8.39	0.87	0.12				computeSpinorProducts
6.29	0.96	0.09				computeME2
4.90	1.03	0.07	12000000	5.83	5.83	ocl_rng_rand48_getDouble
4.20	1.09	0.06	49579320	1.21	1.21	gsl_complex_mul
3.50	1.14	0.05	4249656	11.77	14.12	gsl_complex_div
2.80	1.18	0.04	2833104	14.12	14.12	gsl_complex_add
2.10	1.21	0.03	2124828	14.12	27.13	anomalousAmplitudePMM
2.10	1.24	0.03				generateRambo
2.10	1.27	0.03	25666208	1.17	1.17	gsl_complex_conjugate
1.75	1.30	0.03				computeScalarProducts
1.40	1.32	0.02	2124828	9.41	38.97	QEDAmplitudePMM
1.40	1.34	0.02				selectEvent
1.40	1.36	0.02				updateStatistics
1.05	1.37	0.02				resetME2
0.70	1.38	0.01	26998624	0.37	0.37	gsl_complex_abs2
0.70	1.39	0.01	4249656	2.35	2.35	gsl_complex_abs
0.70	1.40	0.01	2124828	4.71	34.26	QEDAmplitudePPM
0.70	1.41	0.01	708276	14.12	60.22	anomalousAmplitudeMMM
0.70	1.42	0.01	708276	14.12	60.22	anomalousAmplitudePPP
0.70	1.43	0.01				gsl_complex_mul_imag
0.00	1.43	0.00	2124828	0.00	13.01	anomalousAmplitudePPM
0.00	1.43	0.00	1	0.00	0.00	_init
0.00	1.43	0.00	1	0.00	0.00	initSpinorProducts
0.00	1.43	0.00	1	0.00	0.00	main

Call Graph

```
index % time    self  children    called     name
                                             <spontaneous>
[1]    100.0    0.02    1.84                 main [1]
                0.28    0.71  708276/708276      computeME2 [2]
                0.17    0.22 1000000/1000000     computeSpinorProducts [3]
                0.11    0.13 1000000/1000000     generateRambo [5]
                0.07    0.00 1000000/1000000     selectEvent [14]
                0.06    0.01 1000000/1000000     computeScalarProducts [15]
                0.05    0.00  708276/708276      updateStatistics [18]
                0.03    0.00 1000000/1000000     sortPhotons [22]
                0.01    0.00 1000000/1000000     resetME2 [25]
                0.00    0.00       2/4           getticks [27]
                0.00    0.00       1/1           getTicksPerSecond [29]
              0.00    0.00       1/1         initPhysicalParameters(PhysicalParameters_t*, int) [107]
                0.00    0.00       1/1           initEe3p [30]
                0.00    0.00       1/1           ocl_rng_rand48_setSeed [34]
                0.00    0.00       1/1           initRambo [31]
                0.00    0.00       1/1           initStatistics [33]
                0.00    0.00       1/1           finalizeStatistics [28]
            0.00    0.00       1/1         writeResults(PhysicalParameters_t const*, Statistics_t const*, double) [
```

-O3 -march=native -funroll-loops

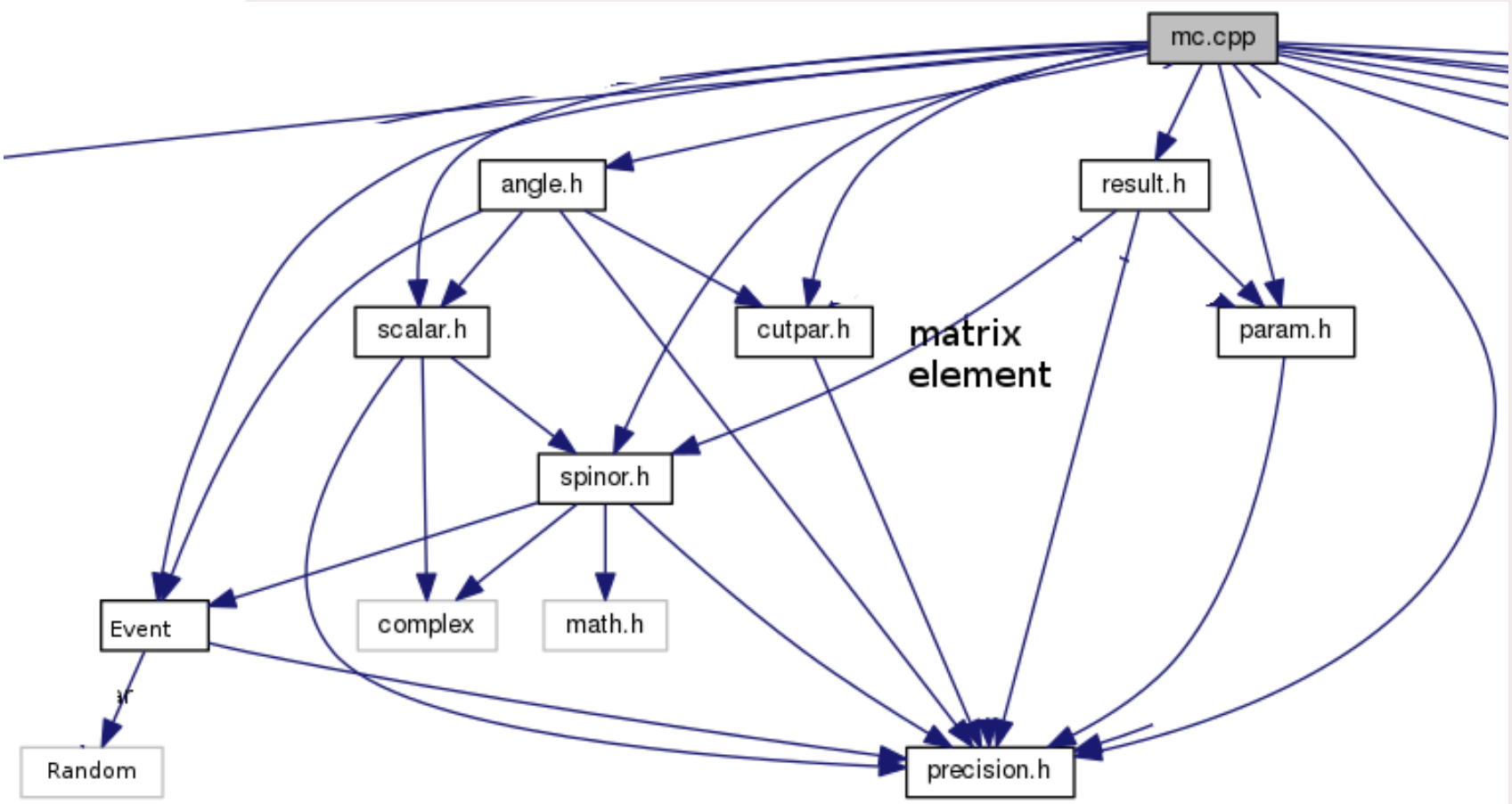
```
index % time    self  children    called     name
                                                 <spontaneous>
[1]     44.8    0.12    0.52                 computeSpinorProducts [1]
                0.19    0.00 20000000/46914488     gsl_complex_mul_real [3]
                0.16    0.00 20000000/20000000     gsl_complex_negative [4]
                0.15    0.00 10000000/10000000     gsl_complex_sub [5]
                0.02    0.00 20000000/25666208     gsl_complex_conjugate [16]
-----------------------------------------------
                                                 <spontaneous>
[2]     41.6    0.09    0.51                 computeME2 [2]
                0.16    0.00 16998624/46914488     gsl_complex_mul_real [3]
                0.02    0.06 2124828/2124828     QEDAmplitudePMM [7]
                0.01    0.06 2124828/2124828     QEDAmplitudePPM [8]
                0.03    0.03 2124828/2124828     anomalousAmplitudePMM [12]
                0.01    0.03  708276/708276      anomalousAmplitudePPP [14]
                0.01    0.03  708276/708276      anomalousAmplitudeMMM [13]
                0.00    0.03 2124828/2124828     anomalousAmplitudePPM [18]
                0.01    0.00 5666208/49579320     gsl_complex_mul [10]
                0.01    0.00 5666208/25666208     gsl_complex_conjugate [16]
                0.01    0.00 16998624/26998624     gsl_complex_abs2 [22]
-----------------------------------------------
                0.01    0.00  708276/46914488     anomalousAmplitudePPP [14]
                0.01    0.00  708276/46914488     anomalousAmplitudeMMM [13]
                0.02    0.00 2124828/46914488     QEDAmplitudePPM [8]
                0.02    0.00 2124828/46914488     QEDAmplitudePMM [7]
                0.02    0.00 2124828/46914488     anomalousAmplitudePPM [18]
                0.02    0.00 2124828/46914488     anomalousAmplitudePMM [12]
                0.16    0.00 16998624/46914488     computeME2 [2]
                0.19    0.00 20000000/46914488     computeSpinorProducts [1]
[3]     30.8    0.44    0.00 46914488          gsl_complex_mul_real [3]
-----------------------------------------------
                0.16    0.00 20000000/20000000     computeSpinorProducts [1]
[4]     11.2    0.16    0.00 20000000          gsl_complex_negative [4]
```

Index by function name

Boucle principale

```
//!>  Start of integration loop
startTime = getticks ();
int ev, evSelected = 0;
double evWeight;
for (ev=0; ev < run->nbrOfEvents; ev++) {

  // Reset Matrix elements
  resetME2( &ee3p );

  // Event generator
  evWeight = generateRambo (&rambo, outParticles, 3, run->ETotal);
  evWeight = evWeight * run->cstVolume;

  // Sort outgoing photons by energy
  sortPhotons (outParticles);

  // Spinor inner product, scalar product and
  // center-of-mass frame angles computation
  computeSpinorProducts (&ee3p.spinor, ee3p.momenta); // intermediate computation
  computeScalarProducts (&ee3p);

  if (selectEvent (&pParameters, &ee3p)) {
    computeME2 (&ee3p, &pParameters, run->ETotal);
    updateStatistics (&statistics, &pParameters, &ee3p, evWeight);
    evSelected++;
  }
}
```

Diagramme


mc.cpp
angle.h
result.h
scalar.h
cutpar.h
matrix
element
param.h
spinor.h
Event
complex
math.h
Random
precision.h

INSTITUT DE PHYSIQUE NUCLÉAIRE
ORSAY

concurrence