Indirect and direct ray tracing for the RICH detector - the basic formalisms

Dirk De Schepper
Argonne National Laboratory

October 24, 1997

Abstract

In this report the mathematical derivation of the formulae used in the direct and indirect ray tracing routines is presented. The main object is to document the reconstruction software being developed for the RICH detector.

1 Indirect Ray Tracing (IRT)

Formulation of the physical problem

Given a track and a hit in the RICH photodetector plane, we want to know under what angle the photon was emitted. Depending on whether the photon comes from the aerogel or from the gas, we can estimate from which point it was emitted. Since we do not know where it was emitted from, we will reconstruct the angle for both cases and "see which one makes sense".

A note on the notation. Throughout this report points are identified by capital letters. The vector in the hermes coordinate frame is represented by the same capital letter, with an arrow over it. All other vectors are represented by the capital letters identifying the start and the end point of the vector, with an arrow over the vector.

The geometrical problem can be formulated as follows using the terminology of figure 1. Given point E, the likely emission point, point D, the

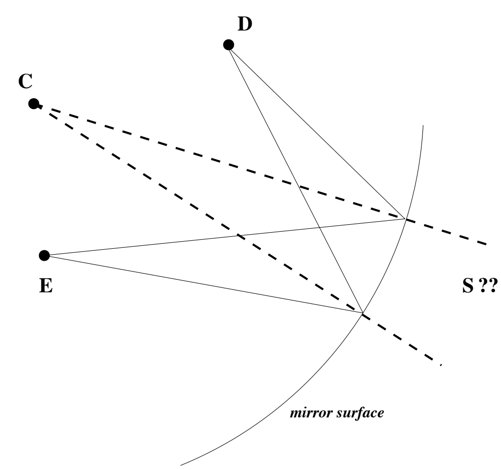

Figure 1: The problem of indirect ray tracing

detection point and C, the center of the spherical mirror the photon scatters from, find the point S on the surface of the mirror where the photon scattered.

The properties of point S are threefold, and are here formulated in vector language:

1. $\vec{CS}$ is coplanar with $\vec{CE}$ and $\vec{CD}$.

2. $|\vec{CS}| = \mathrm{R}$

3. The angle between $\vec{CS}$ and $\vec{CE}$ is the same at between $\vec{CD}$ and $\vec{CS}$

Mathematical formulation of the problem

For the mathematical formulation of the problem it is easier to switch to an euclidean base with C as the origin. The u axis is defined along $\vec{CE}$. The v axis is coplanar with $\vec{CE}$ and $\vec{CD}$, and oriented such that $\hat{v} \cdot \vec{CD} > 0$. We used the usual *caret* notation to indicate a unit vector. The third axis is defined such that base defines a righ-handed coordinate system. We can then explicitly write down the components of the vectors. The vectors become:

$$
\begin{array}{lcl}
\vec{CE} & = & (\quad a \quad , \quad 0 \quad , \quad 0 \quad) \\
\vec{CD} & = & (\quad d*\cos(\theta) \quad , \quad d*\sin(\theta) \quad , \quad 0 \quad) \\
\vec{CS} & = & (\quad R*\cos(\phi) \quad , \quad R*\sin(\phi) \quad , \quad 0 \quad)
\end{array}
$$

Hence $|\vec{CE}| = \text{a}$ and $|\vec{CD}| = \text{d}$. The angles are defined in figure 2. Clearly the following holds:

- $a, d, R > 0$
- $0 < \theta < \pi$
- $0 < \phi < \theta$

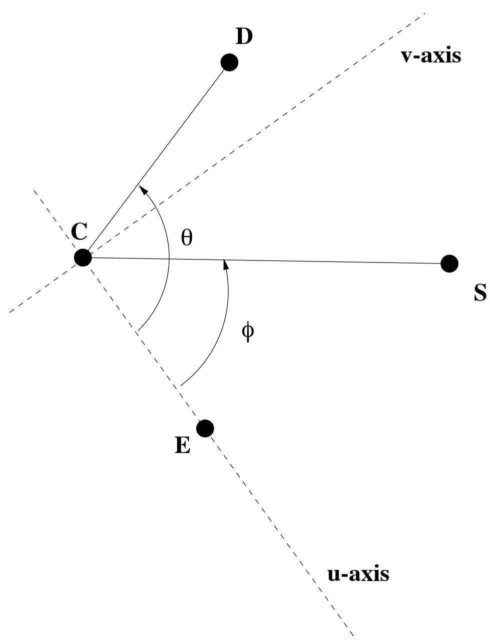

Figure 2: Definition of the base and the angles

The coplanarity requirement has been satisfied by the choice of the axes. The parametrization of S ensures that it will be on the mirror surface. The remaining parameter ϕ now needs to be determined from the equality of the incident and reflected angles. This leads to *two* (redundant) equations for the unit vectors:

$$\hat{SC} \cdot \hat{SE} = \hat{SD} \cdot \hat{SC}$$

$$\hat{SC} \times \hat{SE} = \hat{SD} \times \hat{SC}$$

The second equation is a vector equation, but since all three vectors are coplanar, the only parameter is the length of the vector product.

The essential vectors in these equations can be written in our euclidean base as:

$$\begin{array}{lclrcrcc} \vec{SC} & = & (& \text{-R*cos}(\phi) & , & \text{-R*sin}(\phi) & , & 0 \ ) \\ \vec{SE} & = & (& \text{a-R*cos}(\phi) & , & \text{-R*sin}(\phi) & , & 0 \ ) \\ \vec{SD} & = & (& \text{d*cos}(\theta)\text{-R*cos}(\phi) & , & \text{d*sin}(\theta)\text{-R*sin}(\phi) & , & 0 \ ) \end{array}$$

And hence the equality of the cosines of the angles can be explicitly written as:

$$\frac{(cos(\phi), sin(\phi)) \cdot (a - R * cos(\phi), -R * sin(\phi))}{|\vec{SE}|} =$$
$$\frac{(cos(\phi), sin(\phi)) \cdot (d * cos(\theta) - R * cos(\phi), d * sin(\theta) - R * sin(\phi))}{|\vec{SD}|}$$

Which reduces easily to:

$$(a * cos(\phi) - R) * |\vec{SD}| = (d * (cos(\theta) * cos(\phi) + sin(\theta) * sin(\phi)) - R) * |\vec{SE}|$$

The vector equation represents the equality of the sinuses of the angles, and has only a non-zero component along the w-axes. This component is:

$$(R * cos(\phi) * sin(\phi) + sin(\phi) * (a - R * cos(\phi))) * |\vec{SD}| \quad =$$
$$((d * cos(\theta) - R * cos(\phi)) * (-sin(\phi)) + cos(\phi) * (d * sin(\theta) - R * sin(\phi))) * |\vec{SE}|$$

Which reduces to:

$$(a * sin(\phi)) * |\vec{SD}| = d * (-cos(\theta) * sin(\phi) + sin(\theta) * cos(\phi)) * |\vec{SE}|$$

To avoid the square roots in the absolute norms, we take the ratio of the two equations:

$$\frac{(a*cos(\phi)-R)}{a*sin(\phi)}=\frac{(d*(cos(\theta)*cos(\phi)+sin(\theta)*sin(\phi)-R))}{d*(sin(\theta)*cos(\phi)-cos(\theta)*sin(\phi))}$$

or, transferring terms:

$$\begin{aligned} & (a*cos(\phi)-R)*(d*(sin(\theta)*cos(\phi)-cos(\theta)*sin(\phi))) \\ = & (a*sin(\phi))*(d*(cos(\theta)*cos(\phi)+sin(\theta)*sin(\phi)-R)) \end{aligned}$$

Using basic goniometric relations this equation can be rewritten as:

$$a*d*sin(\theta-2*\phi)+R*(a*sin(\phi)-d*sin(\theta-\phi))=0$$

The derivative of the function on the left hand side is:

$$-2*a*d*cos(\theta-2*\phi)+R*(a*cos(\phi)+d*cos(\theta-\phi))$$

The solution to the equation can now be found from Newton-Raphson iterations. Since the result will be the angle, no further work is necessary. Compared to the procedure reported by R. Forty in the CERN report LHC-B/96-5, this method requires less calculations and yields a well-behaved iterative problem that converges in one or two iterations. Its derivation is - in my opinion - also much simpler.

The reflection point on the mirror can now be found as follows:

$$\begin{aligned} \vec{S} &= \vec{C}+R*cos(\phi)*\hat{u}+R*sin(\phi)*\hat{v} \\ \hat{u} &= \frac{\vec{CE}}{a} \\ \hat{v} &= \frac{\vec{CD}-d*cos(\theta)*\hat{u}}{d*sin(\theta)} \end{aligned}$$

which can be combined to give the explicit result:

$$\vec{S} = \vec{C}+\left(\frac{R*cos(\phi)}{a}-\frac{R*sin(\phi)*cos(\theta)}{sin(\theta)}\right)*\vec{CE}+\frac{R*sin(\phi)}{d*sin(\theta)}*\vec{CD}$$

2 Direct Ray Tracing (DRT)

The direct ray tracing problem can be formulated as follows: given the emission point (E) and the momentum vector ($\vec{P}$), determine the point S where the ray reflects from the surface and the point D where the ray hits the detector. For the purpose of constructing the cerenkov ellips and/or its center, it is sufficient to determine only $\vec{D}$. However, it is also necessary to determine the angle of incidence on the detector (to calculate the acceptance in the Winston cone). To this end we can return $\vec{S}$ or $\vec{SD}$.

A sketch illustrating the problem, as well as the definitions we will use in the further derivation, is shown in figure 3.

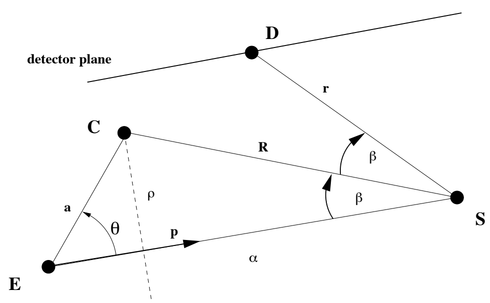

Figure 3: Sketch of the direct ray tracing problem

ρ is the perpendicular distance of C to the EC line. It follows immediately that:

$$\begin{aligned} \rho &= a * sin(\theta) \\ &= R * sin(\beta) \\ \alpha &\equiv |ES| \\ &= a * cos(\theta) + R * cos(\beta) \end{aligned}$$

Since ρ can be eliminated we have β immediately. α is then also directly calculable. This defines the reflection point S and the angle under which the ray exits from S.

We now define a (u,v) coordinate system as follows. The coordinate system has its origin in the point E. The u-axis will be along the momentum vector $\vec{P}$. The v-axis is perpendicular to that, in the ECS plane, and such that the v-component of C is positive. Explicitly we can describe the basis:

$$\hat{u} = \frac{\vec{P}}{|\vec{P}|}$$
$$\hat{v} = \frac{\vec{EC} - a * cos(\theta) * \hat{u}}{a * sin(\theta)}$$

This allows us to write:

$$\vec{ES} = (\alpha, 0, 0)$$
$$\vec{ED} = \vec{ES} + r * (-cos(2 * \beta), sin(2 * \beta), 0)$$

And using the definition of the base vectors we find:

$$\vec{ED} = \left(\alpha - r * cos(2 * \beta) - \frac{sin(2 * \beta) * cos(\theta)}{sin(\theta)}\right) * \frac{\vec{P}}{|\vec{P}|}$$
$$+ \quad r * sin(2 * \beta) * \frac{\vec{EC}}{a * sin(\theta)}$$
$$\vec{D} = \vec{E} + \left(\alpha - r * cos(2 * \beta) - \frac{sin(2 * \beta) * cos(\theta)}{sin(\theta)}\right) * \frac{\vec{P}}{|\vec{P}|}$$
$$+ \quad r * sin(2 * \beta) * \frac{\vec{EC}}{a * sin(\theta)}$$

The only variable that has not been determined in this expression is r. This variable can be solved from the requirement that D lies on the detector plane. The plane is described by the vector equation:

$$\vec{N} \cdot \vec{D} = c$$

where $\vec{N}$ is a vector perpendicular to the plane and c is a constant that can be determined by multiplying any vector that is know to be in the detector plane with $\vec{N}$. We now have an equation in r. If we define the tilt angle γ of the detector plane as the angle between the y-axis and $\vec{N}$, we can write explicitly:

$$\vec{N} = (0, cos(\gamma), -sin(\gamma))$$

We then find for the equation for r:

$$\begin{aligned} c &= \vec{E} \cdot \vec{N} + \left(\alpha - r * cos(2*\beta) - \frac{sin(2*\beta)*cos(\theta)}{sin(\theta)} \right) * \frac{\vec{P} \cdot \vec{N}}{|\vec{P}|} \\ &+ r * sin(2*\beta) * \frac{\vec{EC} \cdot \vec{N}}{a * sin(\theta)} \end{aligned}$$

The solution for r is given by:

$$r = \frac{c - \vec{E} \cdot \vec{N} - \left(\alpha - \frac{sin(2*\beta)*cos(\theta)}{sin(\theta)} \right) * \frac{\vec{P}\cdot\vec{N}}{|\vec{P}|}}{\frac{sin(2*\beta)*\vec{EC}\cdot\vec{N}}{a*sin(\theta)} - cos(2*\beta) * \frac{\vec{P}\cdot\vec{N}}{|\vec{P}|}}$$

If S has been calculated intermediately, the solution for r can be obtained easier from:

$$\begin{aligned} c &= (\vec{S} + r * \hat{SD}) \cdot \vec{N} \\ r &= \frac{c - \vec{S} \cdot \vec{N}}{\hat{SD} \cdot \vec{N}} \end{aligned}$$

The procedure therefore is to calculate a and θ, then deduce β and α. Then calculate $\frac{\vec{P}\cdot\vec{N}}{|\vec{P}|}$, $\vec{E} \cdot \vec{N}$ and $\vec{EC} \cdot \vec{N}$, which allows the calculation of r. r is then used to determine D.

3 Implementation

IRT

The comments in the code keep track of the floating operations, not integer additions (increments for instance) and logical comparisons. Since this is lifted from the actual code there are some functions that are not defined here. In particular all the vector functions are defined elsewhere. Since they are well-known, however, this should not be a problem in understanding the code.

The average number of iterations required for convergence is 2.3. This means that on the average we need two square roots, 50 additions, 72 multiplications and 20 goniometric functions.

```
/*-----------------------------------------------------------------*\
  i_raytrace: perform indirect ray tracing
\*-----------------------------------------------------------------*/
/* input : emission point(E), hit in detector(D),
           pointer to int to return success
   output: vector S (reflection point), success = 0 or 1
   */


VECT i_raytrace(VECT E, VECT D, int *success)
{
  VECT C, S, CE, CD;
  int i,iWhere;
  double th, a, d;
  double x, dx;
  static const double eps = 1e-6;   /* Desired precision for reconstruction
       angle iteration */

  iWhere = ( D.y < 0.0 ); /* which detector half are we working in ? */
  C = Cm[iWhere];         /* vector to the center of the mirror in this
     detector half */

  CE=vsub(E,C);    /* vector from emission point to mirror center */
  CD=vsub(D,C);           /* vector from detection point to mirror center */
  a=vabs(CE);                   /* emission point to mirror center */
```

```
  d=vabs(CD);                   /* detection point to mirror center */
  th=acos(vdot(CD,CE)/a/d);     /*opening angle in E,C,Hit */

  /* the above calculations involve 15 additions, 11 multiplications
     and two square roots */

  /* Newton-Raphson iteration */
  /* initialise variables (requires 8 additions, 15 multiplications and
     6 goniometric functions */

  i = 0;
  x = th/2.;      /*x is the angle of CS in the (u,v) coordinate system */
  dx = -newp(x,th,a,d,Rm)/newpp(x,th,a,d,Rm);

  while( fabs(dx) > eps && i < NMAX) {
    x += dx;
    dx = -newp(x,th,a,d,Rm)/newpp(x,th,a,d,Rm);
    i++;
    /* every iteration involves 14 multiplications, 9 additions and 6
       goniometric functions */
  }

  if ( i >= NMAX ) {
    fprintf (stdout, "i_raytrace did not converge ! residual = %f\n",dx);
/* not converged */
    *success=0;
  } else {
    *success=1;
  }

  /* Find the reflection point on the mirror */
  /* this implies 13 multiplications and 6 additions */

  S = vadd(C,
   vadd( vmul(Rm*(cos(x)/a-sin(x)/tan(th)/a),CE),
 vmul((Rm*sin(x)/(d*sin(th))),CD)));

  return S;
}
```

```
/*---------------------------------------------------------------------*\
  newp : function used in iterative solution of indirect ray tracing
         problem - see report
\*---------------------------------------------------------------------*/

double newp( double x,double th,double a,double d,double R)
{
  double y;

  y = R*(a*sin(x) - d*sin(th-x)) + a*d*sin(th-2*x);

  return y;
}

/*---------------------------------------------------------------------*\
  newpp : derivative of newp
\*---------------------------------------------------------------------*/

double newpp( double x,double th,double a,double d,double R)
{
  double y;

  y = R*(a*cos(x) + d*cos(th-x)) - 2*a*d*cos(th-2*x);

  return y;
}
```

DRT

The total number of operations in the d_raytrace routine is currently 28 additions, 54 multiplications, 2 square roots and 10 goniometric functions. This means we would expect it to take somewhat more than half the execution time of i_raytrace.

```
/*-------------------------------------------------------------------*\
  d_raytrace: perform direct ray tracing.
              E: emission point
      EP: direction of momentum (notation E is to stress that it is
          a vector relative to point E)
      returns D: hit location on the detector
              SD: vector from reflection point on mirror to D
\*-------------------------------------------------------------------*/

int d_raytrace(VECT E, VECT EP, VECT *D, VECT *SD)
{
  int success,iWhere;
  double a, p, theta;
  double beta, alpha;
  double r;
  double pn, tmp;
  VECT EC, ED, ES, C, S, SDu;
  VECT CS;

  iWhere = ( E.y < 0.0 );
  C = Cm[iWhere];
  EC= vsub(C,E);

  /* determine intersection point with mirror */
  /* calculate a and theta */
  a = vabs(EC);
  p = vabs(EP);
  tmp = vdot(EP,EC)/p/a;

  if (fabs(tmp)<0.000001) {
    return 0;
  }
  theta = acos(tmp);
```

```
  /* deduce alpha and beta */
  beta  = asin(a*sin(theta)/Rm);
  alpha = a*cos(theta)+Rm*cos(beta);

  /* calculate ES and S */
  ES = vmul(alpha/p,EP);
  S  = vadd(E,ES);

  /* determine direction of the vector SD */
  SDu = vunit(vadd(vmul(-(cos(2.*beta)+sin(2.*beta)/tan(theta)),vunit(EP)),
     vmul(sin(2.*beta)/a/sin(theta),EC)));

  /* determine how large SD is */
  r = (det_c[iWhere]-vdot(S,det_N[iWhere]))/(vdot(SDu,det_N[iWhere]));

  /* determine intersection point with detector plate */

  *SD = vmul(r,SDu);
  *D =  vadd (S,*SD);

  return 1;
}
```