Subject: Re: Position calculations on start/stop scintillators
Posted by thuyuk on Thu, 14 May 2015 10:07:46 GMT
View Forum Message <> Reply to Message

Hi Michael,

I'm sorry if I'm being a meathead, but in the following part of the code, it seemed to me that you are doing more or less what you described, am I wrong?

```
 if (time_computed == false || parameter(always_compute_position) != 0)
 {
  // We haven't been informed about the position of the particle impact
  // position or we want to compute the particle position in any case.
  // It will be determined by the following minimum search algorithm.
  double radius_sqr = parameter(radius)*parameter(radius);

  // Tiny random jitter around the beam axis is the starting point.
  x_particle = 0.2*parameter(radius)*(rand()%2000 - 1000)/1000.0;
  y_particle = 0.2*parameter(radius)*(rand()%2000 - 1000)/1000.0;

  // Select the start step size more or less arbitrary.
  // 1/7 of the disk radius seems to be a reasonable choice.
  double stepsize = parameter(radius)/7;
  int direction = 0;
  double last_variance = analyze(x_particle, y_particle, true);
  for (int n = 0;; ++n)
  {
   if (x_particle*x_particle + y_particle*y_particle > 4*radius_sqr) // we left the circular
membrane
   {
    //std::cerr << n << " steps to break the circle at position "
    //    << x_particle << " " << y_particle
    //    << "  stepsize was " << stepsize << std::endl;
    return;
   }
   double variance;
   // From the current position go strictly downhill
   // in one of the four directions (+x,-x,+y,-y)
   // on the chi^2 plane.
   if (    (variance = analyze(x_particle + stepsize, y_particle,          true)) < last_variance)
    x_particle += stepsize;
   else if ((variance = analyze(x_particle - stepsize, y_particle,          true)) < last_variance)
    x_particle -= stepsize;
   else if ((variance = analyze(x_particle,          y_particle + stepsize, true)) < last_variance)
    y_particle += stepsize;
   else if ((variance = analyze(x_particle,          y_particle - stepsize, true)) < last_variance)
    y_particle -= stepsize;
   else
   {
    // If a minimum is found and the stepsize is not small
    // enough, reduce the stepsize and restart searching.
```

```
   if (stepsize > 2.0)
   {
    stepsize /= 2;
   }
   else
   {
    // If the stepsize was small enough (less than 2mm
    // seems to be reasonable, because the spacial
    // resolution of the device is approx. 6-7mm)
    // we are done.
    //std::cerr << n << " steps to find the position " << x_particle << " " << y_particle <<
std::endl;
    break;
   }
  }

  last_variance = variance;
 }
```

In this part, you never use an initial position nor calibration, you search the position starting from the beam axis. Then hit position vs. PMT time difference calibration should also work here, right?

Thank you for your patience!

Tayfun