Optimizing the coputation of LOS PIn / POut for a toroidal vessel

Didier VEZINET

26.06.2017

Contents

Chapter 1

Definitions

Let's consider a orthonormal direct cylindrical coordinate system $(O, \underline{e}_R, \underline{e}_\theta, \underline{e}_Z)$ associated to the orthonormal direct cartesian coordinate system $(O, \underline{e}_X, \underline{e}_Y, \underline{e}_Z)$. Let's consider a fraction of a cone C defined by a segment in (R, Z) coordinates: $\underline{A} = \begin{pmatrix} R_A \\ Z_A \end{pmatrix}$ and $\underline{B} = \begin{pmatrix} R_B \\ Z_B \end{pmatrix}$, associated to the normalized directing vector $\underline{v} = \frac{\overline{AB}}{\|\overline{AB}\|}$ of coordinates (v_R, v_Z). We assume that $A \neq B$, so they can't have identical coordinates.

Then, any point M with coordinates (X, Y, Z) or (R, θ, Z) belongs to the fragment of cone C defined by A and B if and only if:

$$\exists q \in [0; 1] / \begin{cases} R - R_A = q(R_B - R_A) \\ Z - Z_A = q(Z_B - Z_A) \end{cases}$$

Now let's consider a LOS L (i.e.: a half-infinite line) defined by a point D and a normalized directing vector u, of respective coordinates (X_D, Y_D, Z_D) or (R_D, θ_D, Z_D) and (u_X, u_Y, u_Z). Then, point M belongs to L if and only if:

$$\exists k \in [0; \infty[/ \underline{DM} = k\underline{u}$$

Chapter 2

Derivation

Let us now consider all intersections between cone C and semi-line L.

$$\exists (q,k) \in [0;1] \times [0;\infty[/ \quad \begin{cases} R - R_A = q(R_B - R_A) \\ Z - Z_A = q(Z_B - Z_A) \\ X - X_D = ku_X \\ Y - Y_D = ku_Y \\ Z - Z_D = ku_Z \end{cases}$$

Which yields (by combining to keep only unknowns q and k):

$$q(Z_B - Z_A) = Z_D - Z_A + ku_Z$$
$$q^2(R_B - R_A)^2 + 2qR_A(R_B - R_A) = \left(k\underline{u}_{//} + \underline{D}_{//}\right)^2 - R_A^2$$

Where we have introduced $R_D = \sqrt{X_D^2 + Y_D^2}$, $\underline{u}_{//} = u_X\underline{e}_X + u_Y\underline{e}_Y$ and $\underline{D}_{//} = X_D\underline{e}_X + Y_D\underline{e}_Y$. We can then derive a decision tree.

Given that the parallelization will take place on the LOS (i.e.: not on the cones which are parts of the vacuum vessel), we will discriminate case based prioritarily on the components of $\underline{u}$ and D. We will detail only the cases which have solutions, in order to make it as clear as possible for implementation of an efficient algorithm. We will also only consider non-tangential solution, as we are looking for entry/exit points.

2.1 $u_Z = 0$: horizontal LOS

Then:

$$\exists (q,k) \in [0;1] \times [0;\infty[/ \quad \begin{cases} R - R_A = q(R_B - R_A) \\ Z_D - Z_A = q(Z_B - Z_A) \\ X - X_D = ku_X \\ Y - Y_D = ku_Y \\ Z = Z_D \end{cases}$$

If $Z_B = Z_A$ (the cone is horizontal too) and:

- $Z_D = Z_A \Rightarrow$ the cone stands in the same plane as the LOS $\Rightarrow$ infinity of solutions, we consider no solutions as this is a limit case with no clearly identified intersection.
- $Z_D \neq Z_A \Rightarrow$ the cone and the LOS stand in different parallel planes $\Rightarrow$ no solution.

Hence, the only derivable solutions suppose that $Z_B \neq Z_A$.

2.1.1 $Z_B \neq Z_A$ (non-horizontal cone)

Then $q = \frac{Z_D - Z_A}{Z_B - Z_A}$. There are acceptable solution only if $q \in [0;1]$. By introducing $C = q^2(R_B - R_A)^2 + 2qR_A(R_B - R_A) + R_A^2$, we have:

$$\left(k\underline{u}_{//} + \underline{D}_{//}\right)^2 - C = 0 \Leftrightarrow k^2\underline{u}_{//}^2 + 2k\underline{u}_{//} \cdot \underline{D}_{//} + \underline{D}_{//}^2 - C = 0$$

Then introducing $\Delta = 4\left(\underline{u}_{//} \cdot \underline{D}_{//}\right)^2 - 4\underline{u}_{//}^2\left(\underline{D}_{//}^2 - C\right) = 4\delta$, there are non-tangential solutions only if $\left(\underline{u}_{//} \cdot \underline{D}_{//}\right)^2 > \underline{u}_{//}^2\left(\underline{D}_{//}^2 - C\right)$. It is necessary to compute the solutions k because we need to check if $k >= 0$.

$$k_{1,2} = \frac{-\underline{u}_{//} \cdot \underline{D}_{//} \pm \sqrt{\delta}}{\underline{u}_{//}^2}$$

Hence, we have solutions if:

$$\begin{cases} u_Z = 0 \\ Z_B \neq Z_A \\ \frac{Z_D - Z_A}{Z_B - Z_A} \in [0;1] \\ k_{1,2} = \frac{-\underline{u}_{//} \cdot \underline{D}_{//} \pm \sqrt{\delta}}{\underline{u}_{//}^2} \geq 0 \end{cases}$$

2.2 $u_Z \neq 0$: non-horizontal LOS

Then $k = q\frac{Z_B - Z_A}{u_Z} - \frac{Z_D - Z_A}{u_Z}$, which means:

$$\begin{aligned} & q^2(R_B - R_A)^2 + 2qR_A(R_B - R_A) + R_A^2 \\ = & \left(\left(q\frac{Z_B - Z_A}{u_Z} - \frac{Z_D - Z_A}{u_Z}\right)\underline{u}_{//} + \underline{D}_{//}\right)^2 \\ = & \left(q\frac{Z_B - Z_A}{u_Z} - \frac{Z_D - Z_A}{u_Z}\right)^2 \underline{u}_{//}^2 + 2\left(q\frac{Z_B - Z_A}{u_Z} - \frac{Z_D - Z_A}{u_Z}\right)\underline{u}_{//} \cdot \underline{D}_{//} + \underline{D}_{//}^2 \\ = & q^2\left(\frac{Z_B - Z_A}{u_Z}\right)^2 \underline{u}_{//}^2 - 2q\frac{Z_B - Z_A}{u_Z}\frac{Z_D - Z_A}{u_Z}\underline{u}_{//}^2 + \left(\frac{Z_D - Z_A}{u_Z}\right)^2 \underline{u}_{//}^2 + 2q\frac{Z_B - Z_A}{u_Z}\underline{u}_{//} \cdot \underline{D}_{//} - 2\frac{Z_D - Z_A}{u_Z}\underline{u}_{//} \cdot \underline{D}_{//} + \underline{D}_{//}^2 \end{aligned}$$

Hence:

$$\begin{aligned} 0 = & \quad q^2\left((R_B - R_A)^2 - \left(\frac{Z_B - Z_A}{u_Z}\right)^2 \underline{u}_{//}^2\right) \\ + & \quad 2q\left(R_A(R_B - R_A) + \frac{Z_B - Z_A}{u_Z}\frac{Z_D - Z_A}{u_Z}\underline{u}_{//}^2 - \frac{Z_B - Z_A}{u_Z}\underline{u}_{//} \cdot \underline{D}_{//}\right) \\ - & \quad \left(\frac{Z_D - Z_A}{u_Z}\right)^2 \underline{u}_{//}^2 + 2\frac{Z_D - Z_A}{u_Z}\underline{u}_{//} \cdot \underline{D}_{//} - \underline{D}_{//}^2 + R_A^2 \end{aligned}$$

We can then introduce:

$$\begin{cases} A = (R_B - R_A)^2 - \left(\frac{Z_B - Z_A}{u_Z}\right)^2 \underline{u}_{//}^2 \\ B = R_A(R_B - R_A) + \frac{Z_B - Z_A}{u_Z}\frac{Z_D - Z_A}{u_Z}\underline{u}_{//}^2 - \frac{Z_B - Z_A}{u_Z}\underline{u}_{//} \cdot \underline{D}_{//} \\ C = -\left(\frac{Z_D - Z_A}{u_Z}\right)^2 \underline{u}_{//}^2 + 2\frac{Z_D - Z_A}{u_Z}\underline{u}_{//} \cdot \underline{D}_{//} - \underline{D}_{//}^2 + R_A^2 \end{cases}$$

Because of the shape of potential solutions, we have to discriminate the case $A = 0$.

2.2.1 $A = 0$: LOS parallel to one of the cone generatrices

Then, because of the shape of the potential solution, we have to discriminate the case $B = 0$. But in this case we have $C = 0$.

- if $C = 0 \Rightarrow$ no condition on q and k, the LOS is included in the cone $\Rightarrow$ we consider no solution
- if $C \neq 0 \Rightarrow$ Impossible, no solution

Only the case $B \neq 0$ is thus relevant.

$B \neq 0$: LOS not included in the cone

Then, there is either one or no solution:

$$\begin{cases} q = -\frac{C}{2B} & \in [0,1] \\ k = q\frac{Z_B - Z_A}{u_Z} - \frac{Z_D - Z_A}{u_Z} & \geq 0 \end{cases}$$

2.2.2 $A \neq 0$: LOS not parallel to a cone generatrix

Then, we only consider cases with two distinct solutions (i.e.: no tangential case):

$$\begin{cases} B^2 > AC & \\ q = \frac{-B \pm \sqrt{B^2 - AC}}{A} & \in [0,1] \\ k = q\frac{Z_B - Z_A}{u_Z} - \frac{Z_D - Z_A}{u_Z} & \geq 0 \end{cases}$$

Chapter 3

Results

3.1 Upgrades

The algorithm originally implemented in ToFu (hereafter called 'original') resorted to two nested **for** loops in Cython (i.e.: in the General_Geom_cy.pyx file). For each semi-line, a **for** loop was run on all the segments consistuting the POly of the provided vessel. In addition to being slow (despite Cython), this 'original' algorithm seemed to return wrong results in some particular cases, due to the fact it had been written in haste and the analytical derivations had not been preserved.

This Note thus includes a full analytical derivation of the formulas to be used, as these formulas are implemented in ToFu in matricial form.

Indeed, using the above derivations, an intermediate algorithm replacing the inner **for** loop (i.e.: the loop on the segments) bya matricial computation is first created. It is called 'Inter' and is used as an intermediate test-bench to make sure it is faster and returns correct results by benchmarking against 'original', including in the identified problematic cases.

This intermediate check being conclusive, a fully matricial algorithm is derived, handling the coputation of PIn and POut for N LOS with a vessel with Ns segments, called 'Multi', thus getting rid of both **for** loops. That new algorithm is successfully tested, and a last version is derived using only flattened arrays and avoiding redundant declarartions of large arrays, to access an ultimate speed gain? That last algorithm is called 'Multi_Flat'.*

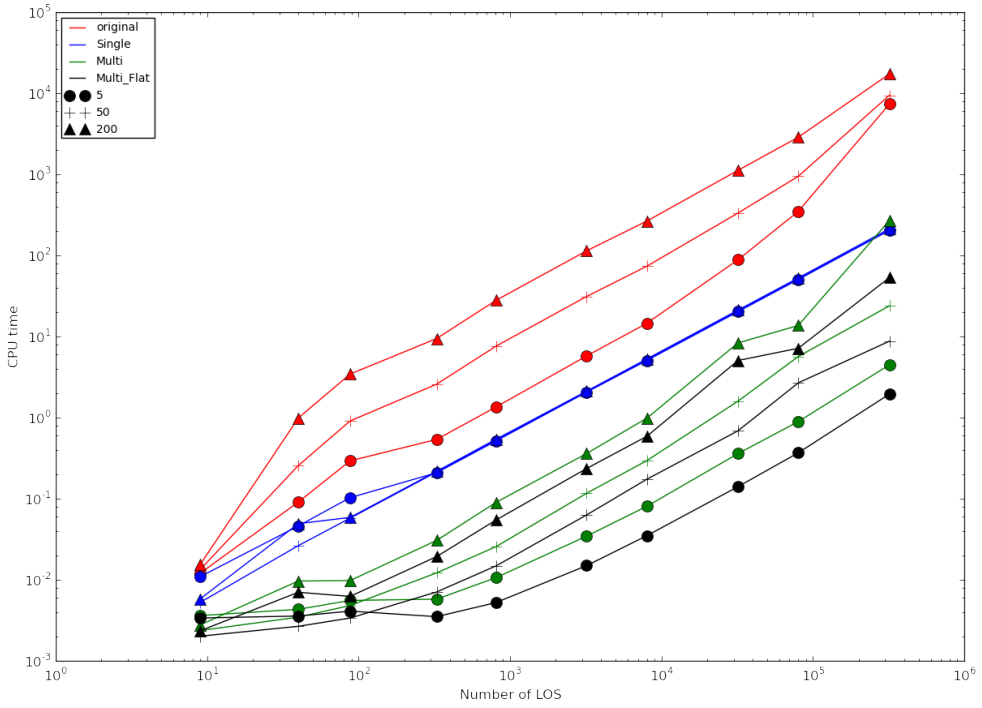
The figure below shows a benchmark on CEA/IRFM/spica of all algorithms ('original', 'Interm', 'Multi' and 'Multi_Flat'), for a several values of N (number of LOS) and several values of Ns (number of segments). Knowing that a typical vessel includes between 50 and 100 segments, and that for a single detector we usually define several tens of thousands of LOS, we can reasonably assume the total speed gain between 'original' and 'Multi_Flat' is typically of the order of .

That upgrade is new in ToFu v.

A test on one of the detectors of the WEST.Bolo diagnostic yielded the following results: computing Detect._set_SinoSpan() took on average 5.7 s with 'Multi_Flat' and 6 min 14 s with 'original', resulting in a speed improvement by a factor 66 approximately.

A step further could b achieved by writting the entire algorithm in pure C/C++ or Fortran and by breaking loops we it becomes clear trhat no solution can be found. The paralization (on the LOS) could be implemented.

Figure 3.1: Benchmark of available algorithms for the computation of PIn / POut for N LOS with a vessel constituted of Ns segments. CPU time is in seconds.



Appendix A

Acceleration radiation from a unique point-like charge

A.1 Retarded time and potential

A.1.1 Retarded time

Deriving the retarded time

Hence $\frac{dR(t_r)}{c} + dt_r = dt$

A.1.2 Retarded potentials

Deriving the potential propagation equations