Quantum Diffusion Element in AT

B. Nash, N. Carmignani

July 24, 2014

Abstract

This document explains the implementation of a quantum diffusion element in the Accelerator Toolbox.

1 Motivation

For the collective effects modeling for the ESRF upgrade phase II, one needs a tracking code that includes impedance effects. As a first step towards this goal, we implement a quantum diffusion element in AT. This will allow single particle tracking to reach the correct zero current equilibrium distribution. The effects of the impedance can then be seen as current dependent perturbations to this.

2 Theory

An electron moves around the ring under the influence of the magnetic fields. One may describe this by a Hamiltonian and the resulting dynamics are symplectic. In addition to this motion, there is also the effect from the synchrotron radiation. This synchrotron radiation has two effects, a damping effect and a diffusion effect.

In the case of linear motion about the closed orbit, the damping effect shows up in the one turn map matrix, M, with the effect that it is no longer symplectic. The eigenvalues now take the form:

$$\lambda_a = e^{i\mu_a + \chi_a} \quad a = 1, 2, 3 \tag{1}$$

where $\mu_a = 2\pi\nu_a$, and ν_a are the tunes and are real in the case of a stable ring. The χ_a are the damping decrements and are real quantities giving the damping rates in units of T_0, the revolution time.

The diffusion results locally from where the radiation occurs. There is a local diffusion matrix and a global diffusion matrix. The global diffusion matrix is related to the local diffusion matrix via (see [1] and [2])

$$\bar{D}(s) = \int_s^{s+C} ds' T_{s' \to s+C} D(s') T_{s' \to s+C}^T \tag{2}$$

where $T_{s' \to s+C}$ is the transfer matrix between s' and $s + C$. The local diffusion

matrix is given as

$$D(s) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d(s) & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

with

$$d(s) = \frac{55}{48\sqrt{3}} \alpha_0 \frac{\gamma^5}{|\rho(s)|^3} \left(\frac{\hbar}{mc} \right)^2 \tag{4}$$

We note that locally the diffusion is only in the energy, but due to dispersion (causing mainly the horizontal diffusion and emittance) and coupling it will be transformed into diffusion in the other planes as well. In AT, the phase space coordinates are given by:

$$\vec{Z} = \begin{pmatrix} x \\ \frac{p_x}{P_0} \\ y \\ \frac{p_y}{P_0} \\ \delta \\ ct \end{pmatrix} \tag{5}$$

The momenta are defined as

$$x' = \frac{p_x}{P_z} \tag{6}$$

with $P_z = P_0(1 + \delta)$.

Given the global diffusion coefficient, the equation for the evolution of the second moment matrix over one turn is

$$\Sigma_2 = M \Sigma_1 M^T + \bar{D} \tag{7}$$

where we recall that M here contains the radiation damping effect. Solving for equilibrium means that the second moments must satisfy

$$\Sigma = M \Sigma M^T + \bar{D} \tag{8}$$

This is the deterministic equation for the moments, but we may also want to include the quantum diffusion in a non-deterministic way, and also to combine it with other effects. In this case, we may use these results to define a random kick each electron receives on each turn. Assuming the processes considered happen over many turns, we assume that lumping the effects into one kick will give the correct dynamics.

To determine the kick given to each electron, we want $\bar{D}$ to be the covariance matrix for the kick. Thus, we simply give a kick that is random but has $\bar{D}$ as covariance. This may be accomplished by performing a Cholesky decomposition on $\bar{D}$. This means, we find the matrix L such that

$$\bar{D} = L^T L \tag{9}$$

Choosing a random vector, each of which elements are Gaussian distributed, with covariance 1, multiplying this vector by L^T, we get the appropriate kick vector.

3 Implementation in AT

We implement a quantum diffusion element modeling the effect on single electrons of the random fluctuations from the photon emission.

To create this element, we have created an element constructor called atQuantDiff. This function takes a ring structure and the name of the element as arguments. It then computes the diffusion matrix for the ring, applies the Cholesky decomposition, and then creates the element above. The diffusion matrix is computed using functions already written for AT. In particular, we use findmpoleraddiffmatrix which computes the local diffusion matrix in multipole elements. These local diffusion matrices are accumulated with the transfer matrix via equation (2). This was already implemented in the function ohmienvelope, but we have extracted the component for the global diffusion matrix, since this is all that is needed. The results of applying this quantum diffusion element on many particles over several damping times, should reproduce a distribution with second moments equivalent to the solution given by the ohmienvelope function.

In the case of an ideal lattice with no coupling or vertical dispersion, the vertical diffusion is 0, and the diffusion matrix is not positive definite which would cause the Cholesky decomposition to fail. In this case, we remove the vertical direction, and do the decomposition just on the horizontal and longitudinal (4-D phase space). We then insert zeros to add the vertical dimension back so the matrix L is 6x6.

To do this, we create a new element pass method called 'QuantDiffPass'. We have written an element creation function called atQuantDiff that computes the diffusion matrix and then L^T from the ring. The output is an element to be inserted into the ring structure.

```
qdelem=atQuantDiff('qdelem',esrf)

qdelem =

       FamName: 'qdelem'
    PassMethod: 'QuantDiffPass'
         Class: 'QuantDiff'
         Lmatp: [6x6 double]
        Length: 0
```

In this element, FamName is the name of the element, QuantDiffPass refers to the pass method, and Lmatp is the transpose of the Cholesky decomposition of the global diffusion matrix $\bar{D}$.

Note that radiation must be off in the ring used in the constructor. The ring to be used for tracking, however, should have radiation on, so that damping will be included.

As an example, using the ESRF lattice, we find a diffusion matrix:

$$\text{diff} = 10^{-8} \times \begin{pmatrix} 0.0186 & 0.0000 & 0 & 0 & 0.0482 & 0.0036 \\ 0.0000 & 0.0000 & 0 & 0 & 0.0000 & -0.0000 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0482 & 0.0000 & 0 & 0 & 0.3642 & 0.0271 \\ 0.0036 & -0.0000 & 0 & 0 & 0.0271 & 0.0027 \end{pmatrix} \tag{10}$$

And the resulting L^T is given by

$$L^T = 10^{-4} \times \begin{pmatrix} 0.1365 & 0 & 0 & 0 & 0 & 0 \\ 0.0000 & 0.0029 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.3530 & 0.0023 & 0 & 0 & 0.4895 & 0 \\ 0.0262 & -0.0003 & 0 & 0 & 0.0365 & 0.0261 \end{pmatrix} \tag{11}$$

4 QuantDiffPass Pass method

The pass method is written in c, with PassFunction for interface to atpass and MexFunction for interface to Matlab. The pass method produces 6 random numbers [1], Gaussian distributed, with $\sigma = 1$ and average 0. Let these numbers be r_i $i = 1 \ldots 6$. Let $\vec{Z}_0$ be the incoming phase space point. The output of the pass method is then simply

$$\vec{Z}_1 = \vec{Z}_0 + L^T \vec{r} \tag{12}$$

We set the seed according to the computer clock (requesting microsecond precision) in the case that we call this for the first turn. For the following turns, we do not reset the seed. This allows for different seeds when the pass method is parallelized, running in different processes.

5 Testing and verification of element

We test the quantum diffusion element by creating an initial distribution of particles, and tracking with this element turned on over several damping times.

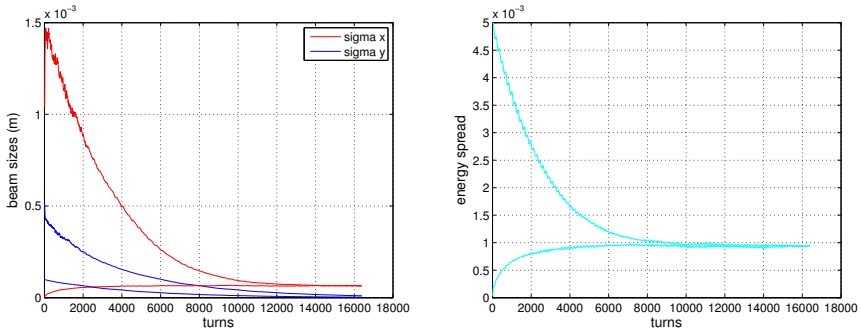
Table 1: Parameters for the S28A lattice. Beta functions are given for the center of the injection straight section.

Parameter	value
C	843.979 m
T_0	2.815 μs
τ_x	8.50 ms = 3020 turns
τ_y	13.0 ms = 4618 turns
τ_z	8.84 ms = 3140 turns
ϵ_x	147 pm
σ_δ	0.947×10^{-3}
β_x	28.0 m
β_y	1.63 m
β_z	3.74 m

As a first example, we take the ideal S28A lattice (no coupling), and use 2048 particles. These are tracked for 16,384 turns. We consider two separate sets of

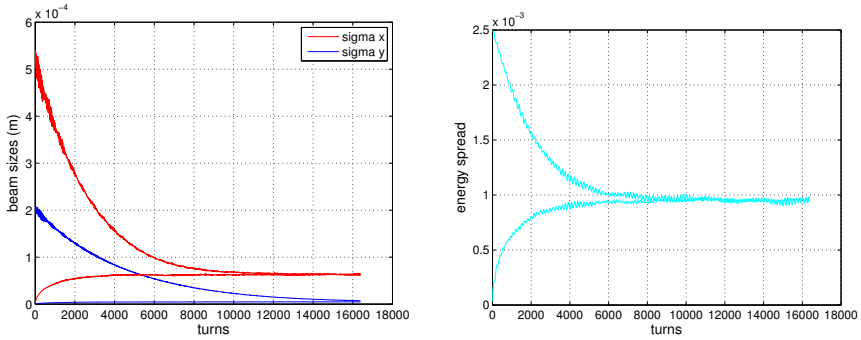
[1] using the function rand() in the standard C library stdlib.h, for uniform distribution, then transformation to a Gaussian

initial conditions, just to check to see that both result in the same equilibrium distribution. For the first initial conditions, we put all the particles on the closed orbit in the horizontal and longitudinal, but an initial vertical emittance of 6.1 nm. For the second initial conditions, we have $\varepsilon_x = 35.7nm$, $\varepsilon_y = 153nm$, and $\varepsilon_z = 9.35\mu m$.



(a) (b)

Figure 1: Evolution of beam sizes (a) and energy spread (b) with quantum diffusion element for uncoupled ideal S28A lattice. Two different sets of initial conditions are used, showing evolution to the same final equilibrium 2048 particles were used and were tracked for 16,384 turns.



(a) (b)

Figure 2: Evolution of beam sizes (a) and energy spread (b) with quantum diffusion element for coupled ideal S28A lattice. Two different sets of initial conditions are used, showing evolution to the same final equilibrium.

6 Comparison of final eigenemittances

The beam sizes following many turns may be compared to the results from the Ohmi-Envelope calculation. In order to compare fewer than all 36 quantities, we

will compute the eigen-emittances. These may be computed from Σ and from the one turn map matrix M as follows. First we compute the normalization matrix A from M such that

$$A^{-1}MA = R \tag{13}$$

with

$$R = \begin{pmatrix} R_x & 0 & 0 \\ 0 & R_y & 0 \\ 0 & 0 & R_z \end{pmatrix} \tag{14}$$

Then we compute the matrix N which should have the form (only if the distribution is exactly a function of the invariants, which is not exactly true)

$$N = -A^T J \Sigma J A = \begin{pmatrix} \epsilon_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \epsilon_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \epsilon_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \epsilon_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \epsilon_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & \epsilon_3 \end{pmatrix} \tag{15}$$

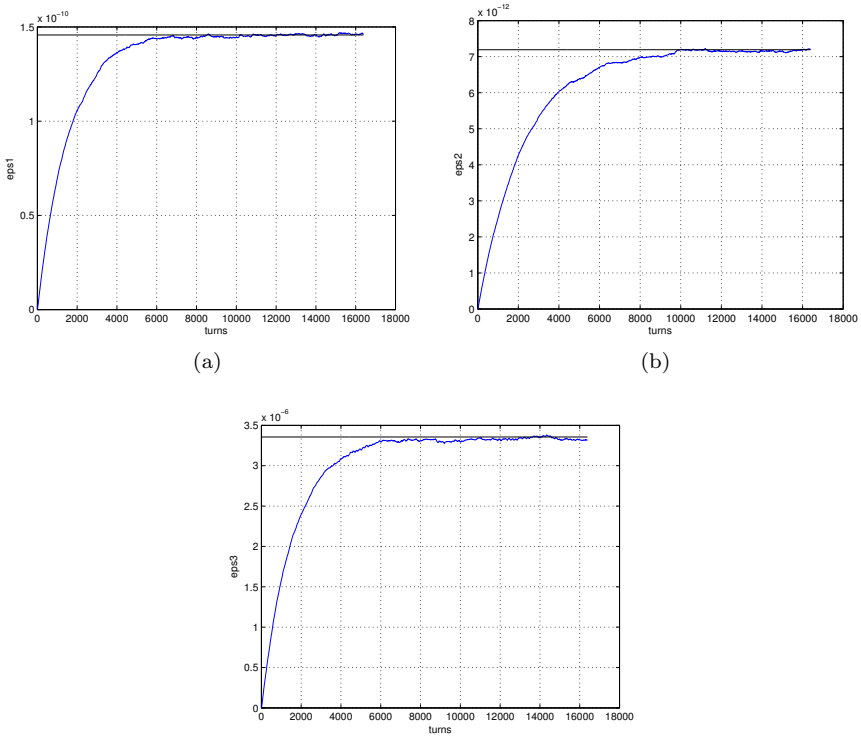


Figure 3: Evolution of eigenemittances versus value computed with Ohmi Envelope. 32,768 particles were used and tracked for 16,384 turns.

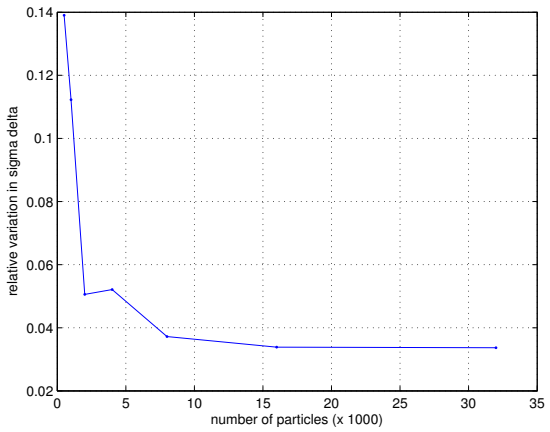


Figure 4: Spread in Energy spread as a function of number of particles in the simulation. Particles were tracked for 16,384 turns and the maximum variation in energy spread was computed over the last 6000 turns.

References

[1] K. Ohmi et. al., 'From the beam-envelope matrix to synchrotron-radiation integrals', Phys. Rev. E 49, 751 (1994)

[2] B. Nash, 'Analytical Approach to Eigen-Emittance Evolution in Storage Rings', Stanford Thesis, SLAC-R820, (2006)