# Advancements in ISR Spectra Modeling with Coulomb Collision Effects

***BRIAN LA ROSA*[1], *DAVID HYSELL*[1], *MARCO MILLA*[2]**

Cornell University [1] , Pontific Catholic University of Peru [2]

**EQUIT-19**

**bl639@cornell.edu**

## 1. INTRODUCTION

- The ISR technique is the most widely used ground-based method for estimating ionospheric plasma parameters, recovered by matching the measured autocorrelation function of the backscattered signal with a theoretical model.
- Based on single-particle statistics and a nonlinear Langevin equation, Milla and Kudeki (2011) developed a new model that incorporates the physics of Coulomb collisions between plasma particles, effects that become significant only at small aspect angles..

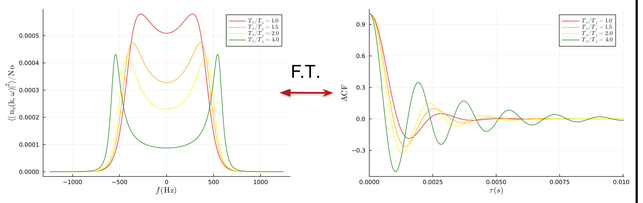


**Figure 1:** Typical ISR spectrum (right) and ACF (left) curves with associated electron temperature dependency.

Motivation:

1. This approach force the use of small time-step discretization generating a large computational time and low efficiency, limiting it use as a regular parameter estimation tool.
2. In this work, we improve the numerical scheme by using higher-order stochastic differential equation solvers.
3. Additionally, Gaussian regression techniques are used to reduce the number of parameter configurations needed to produce the ISR library.

## 2. BACKGROUND

**I) Coulomb Collisions:**

- A fundamental aspect of Coulomb collisions is the long-range nature of the electric force between two charged particles, leading to a broad spectrum of particle deflections.
- Because the effect of these deflections varies for each collision, only a statistical description provides meaningful information about the overall plasma behavior, represented by:

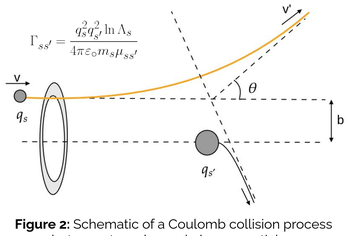


**Figure 2:** Schematic of a Coulomb collision process between two charged plasma particles.

a) Friction Coefficient:

$$A_s(\mathbf{v},t) = \sum_{s'} \Gamma_{ss'} \nabla_{\mathbf{v}} H_{s'}(\mathbf{v})$$

b) Diffusion Coefficient:

$$B_s^2(\mathbf{v},t) = \sum_{s'} \Gamma_{ss'} \frac{\mu_{ss'}}{m_s} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} G_{s'}(\mathbf{v})$$

Here *H* and *G* are known as the Rosenbluth potentials.

## 2.2. LANGEVIN EQUATION

- The Fokker-Planck equation is commonly used to describe the stochastic nature of the Coulomb collisions. Due to its high dimensionality, solving it becomes computationally demanding. Instead, an equivalent mathematical formulation known as the Langevin equation can be used:

**Fokker-Planck Collisional Operator:**

$$\frac{\partial f_s}{\partial t} = -\frac{\partial}{\partial \mathbf{v}}\left[A_s(\mathbf{v},t) f_s(\mathbf{v},t)\right] + \frac{1}{2}\frac{\partial^2}{\partial \mathbf{v}^2}\left[B_s^2(\mathbf{v},t) f_s(\mathbf{v},t)\right]$$

**Complementary perspectives of the same physical process.**

**Langevin Equation:**

$$d\mathbf{v} = A_s(\mathbf{v},t)dt + B_s(\mathbf{v},t)dW + \underbrace{\frac{q_s}{m_s}\mathbf{v} \times \mathbf{B}_\circ dt}_{\text{Gyromotion}}$$

- Although the Fokker-Planck and Langevin equations represent different formulations, they ultimately describe the same underlying dynamics. The critical aspect of this new approach is the inclusion of a Brownian motion term (*dW*).

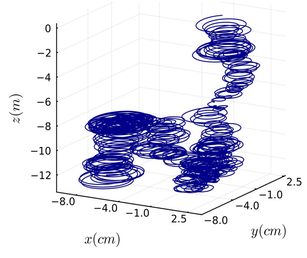


**Figure 3:** A 400 us simulated electron trajectory undergoing friction, diffusion and gyromotion effects.

**The motion is driven by:**

1. Friction: constant loss energy.
2. Diffusion: fluctuating random force.
3. Gyromotion: helical motion around B field z-axis.

## 4. SDE SOLVER

- The previously stated Langevin equation cannot be solved analytically and must be approached numerically. To address this, we employ a higher-order Stochastic Differential Equation (SDE) algorithm that eliminates the need for small time-step discretizations.

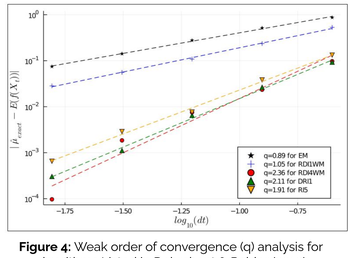


**Figure 4:** Weak order of convergence (q) analysis for algorithms Listed in Debrabant & Robler (2010)

DifferentialEquations.jl

| Algorithm | Description | Features |
|---|---|---|
| EM | Euler-Maruyama | Explicit, Weak order 1 |
| RDI1WM | Stochastic Runge-Kutta based algorithm | Explicit, Weak order 1 |
| RDI4WM | Stochastic Runge-Kutta based algorithm | Explicit, Weak order 2 |
| RI5 | Stochastic Runge-Kutta based algorithm | Explicit, Weak order 2 |
| DRI1 | Approximation EM (Drift Contition) | Explicit, Weak order 1 |

**Table1: :** List of Stochastic Solvers.

- In Figure IV, a weak convergence test shows the relationship between the precision and computational time-step discretization.
- We identify that the RDI4WM algorithm has the fewer error measurement from solvers listed in table I, better than the EM algorithm used by Milla and Kudeki.

## 4.1 RESULTS - MONTE CARLO SIMULATION

- Since the Langevin equation only describes the trajectory of a single particle, we require a large amount of computations to recover the statistics needed for computing the ACF, both in the parallel (l) and perpendicular (p) directions with respect to the B field:

$$\langle e^{i(k_\parallel \Delta l + k_\perp \Delta p)} \rangle = e^{-\frac{1}{2}k_\parallel^2 \langle \Delta l^2 \rangle} \times e^{-\frac{1}{2}k_\perp^2 \langle \Delta p^2 \rangle}$$

**Simulation Details**

- Trajectories: 5000
- dt = 1e-8 (fixed)
- Simulation time: 2.5 ms

**Performance:**

- Exe. Time = 20 min
- Cores: 32
- Mem. Allocation: ~5 Gb Ram

The new algorithm allows a stable evolution of the Langevin equation, with fixed time step.

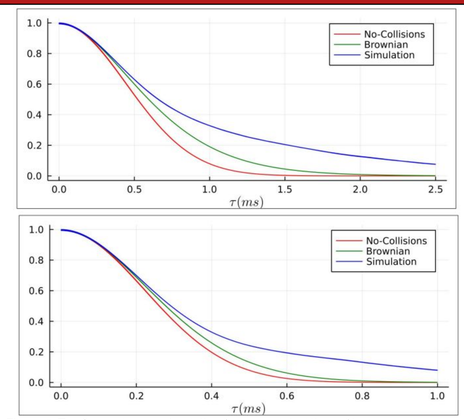


**Figure 5:** Top figure shows a simulation for a correlated time of t = 1 ms and an aspect angle of α = 1.0°. Bottom figure shows a simulation for t = 2.5 ms and aspect angle of α = 0.5°. In both cases we compare with previous models.

## 4.2. RESULTS - GAUSSIAN PROCESS REGRESSION

Radial Basis Function (RBF) Kernel:

$$K_1(x_1,x_2) = \sigma^2 \exp\left(-\frac{||x_1 - x_2||^2}{2l^2}\right)$$

- The advantage of using the Langevin equation lies in its ability to efficiently handle the large number of simulations required to generate the autocorrelation curves. However, each set of plasma parameters produces a distinct ACF, making it computationally expensive to cover the full parameter space. To address this, we employ a specialized interpolation method known as Gaussian Process Regression (GPR).
- GPR relies on a *kernel* function to define the correlation between points in parameter space, allowing smooth interpolation based on limited observations.
- Figure 6 illustrates the implementation of Gaussian Process Regression (GPR) over a three-parameter space (Te. α, τ). The white circles represent simulation data points.
- Using a radial basis function (RBF) kernel, we interpolate the autocorrelation function across the entire parameter space, avoiding the computation of every individual parameter configuration explicitly.

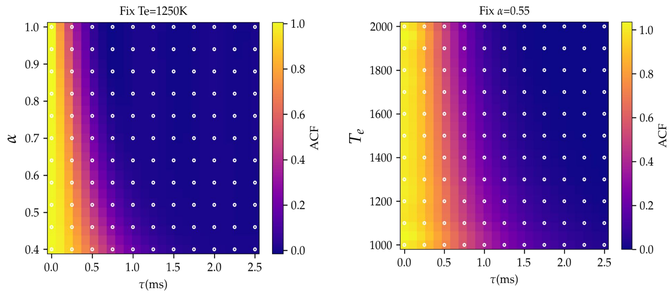


**Figure 6:** Gaussian Regression interpolation for calculating the ACF. Right panel shows the results with constant electron temperature, while left panel maintain a fix magnetic aspect angle.

## 5. CONCLUSIONS

- The RDI4WM higher-order SDE solver have been used to solved the Milla & Kudeki model for the electron particle dynamics, showing a better performance among the other algorithms.
- Numerical instability, caused by the stiffness of the Langevin equation due to large friction and diffusion coefficients, was mitigated by employing a small fixed time step of dt =1e-8.
- To reduce the number of required simulations for exploring the full parameter space, a 3D Gaussian process regression technique was used to recover the autocorrelation function based on the proposed SDE solver.

**REFERENCES:**

[1] Kudeki, E. & Milla, M.A. (2011). IEEE Transactions on Geoscience and Remote. Incoherent scatter spectral theories-Part II: A general result for small magnetic as Sensing, vol 49, pp 315-328. doi: 10.1109/TGRS.2010.2057252

[2] Debrabant &Robler A. (2010)). Mathematics and Computing in simulations. Classification of Stochastic Runge–Kutta Methods for the Weak Approximation of Stochastic Differential Equations. vol 04, pp 408-420, doi: 10.1016/j.matcom.2007.04.016

[3] Platen, Eckhard (1999). An introduction to numerical methods for stochastic differential equations. Acta Numerica, vol 8, pp 197-246. doi: 10.1017/S0962492900002920

[4] Rackauckas, Christopher and Nie, Qing (2017). The Journal of Open Research Software. A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia Vol 5. doi:10.5334/jors.151