PyBoltz – a New Python Swarm Simulation Code

B. Al Atoum, S. Biagi, D. Gonzalez Diaz, B.J.P. Jones and A.D.McDonald

Electron Transport in Gaseous Detectors with a Python-based Monte Carlo Simulation Code

B. Al Atoum[a], S. F. Biagi[b], D. González-Díaz[c], B.J.P. Jones[a], A.D. McDonald[a]

[a] *Department of Physics, University of Texas at Arlington, Arlington, TX 76019, USA*
[b] *University of Liverpool, Physics Department, Liverpool L69 7ZE, United Kingdom*
[c] *Instituto Galego de Física de Altas Enerxías, Univ. de Santiago de Compostela, Campus sur, Rúa Xosé María Suárez Núñez, s/n, Santiago de Compostela, E-15782, Spain*

Abstract

Understanding electron drift and diffusion in gases and gas mixtures is a topic of central importance for the development of modern particle detection instrumentation. The industry-standard `MagBoltz` code has become an invaluable tool during its 20 years of development, providing capability to solve for electron transport ('swarm') properties based on a growing encyclopedia of built-in collision cross sections. We have made a refactorization of this code from `FORTRAN` into `Cython`, and studied a range of gas mixtures of interest in high energy and nuclear physics. The results from the new open source `PyBoltz` package match the outputs from the original `MagBoltz` code, with comparable simulation speed. An extension to the capabilities of the original code is demonstrated, in implementation of a new Modified Effective Range Theory interface. We hope that the versatility afforded by the new `Python` code-base will encourage continued use and development of the `MagBoltz` tools by the particle physics community.

Paper submission to the Computer Physics Communications journal.

Electron Drift and Longitudinal Diffusion in High Pressure Xenon-Helium Gas Mixtures

---

The NEXT Collaboration

A.D. McDonald,[1,a] K. Woodruff,[1] B. Al Atoum,[1] D. González-Díaz,[2] B.J.P. Jones,[1] C. Adams,[10]

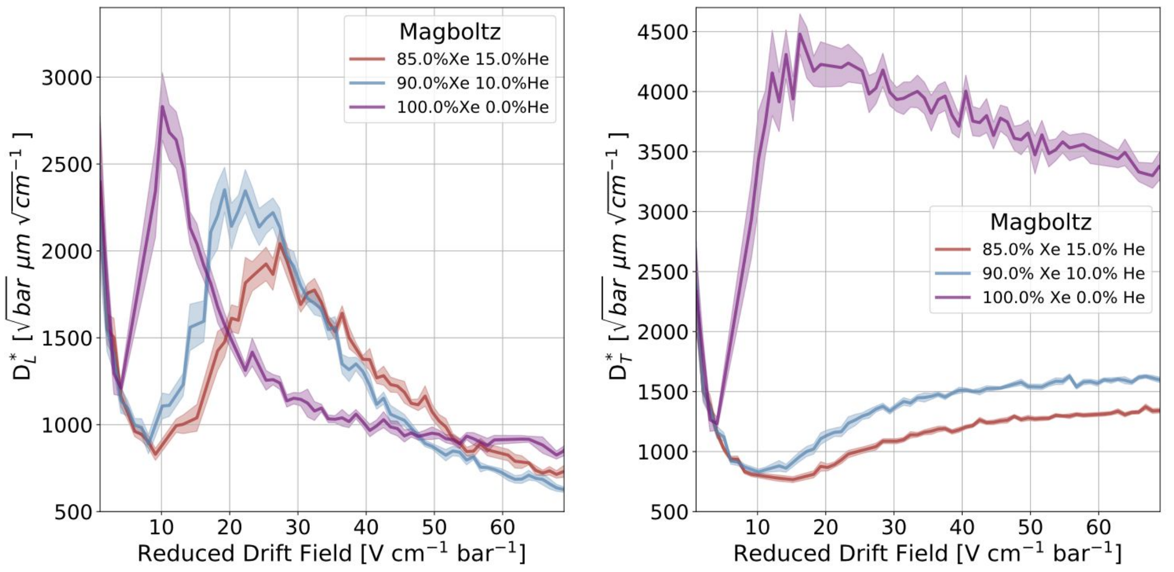


Figure 1. MagBoltz predictions of the dependence of the reduced longitudinal (left) and transverse (right) diffusion coefficients(equation 3.1) on helium concentration. The line represents the exact value from the simulation and the shaded region is the error in the simulation.

JINST 14 (2019) no.08, P08009.

About Magboltz.

- Predicts electron transport properties including velocity, longitudinal and transverse diffusion, attachment, Townsend coefficients, mean energy, for electron swarms drifting in gas mixtures.
- Since ~1999 MagBoltz has been a workhorse code for the field of gas detectors.

The process of Magboltz.

$$\frac{d}{dt}[f_\alpha(\vec{x},\vec{v},t)] \equiv \frac{\partial f_\alpha}{\partial t} + \vec{v}\cdot\nabla_{\vec{x}} f_\alpha + \frac{Z_\alpha e}{m_\alpha}\left(\vec{E}+\vec{v}\times\vec{B}\right)\cdot\nabla_{\vec{v}} f_\alpha = C_\alpha$$

Monte Carlo calculation of electron transport coefficients in counting gas mixtures: II. Mixtures containing neon and carbon dioxide

G.W. Fraser, E. Mathieson

⊞ Show more

https://doi.org/10.1016/0168-9002(86)90418-3 Get rights and content

- Although the name suggests that it uses the boltzmann equation, Magboltz has not been using it since the early releases in 1990s.

- Since 2001, Magboltz used the methods of Frasier and Mathieson, and collision-by-collision simulations.

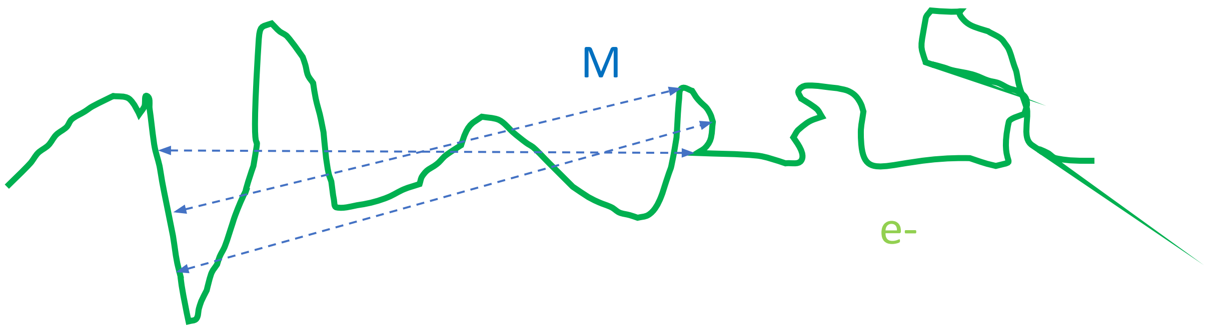
Basics of the Monte Carlo simulation.

$$D_x = \sum_i^{N_{coll}} \frac{(x_i - x_{i-M})^2}{t_i - t_{i-M}} \times \frac{t_i - t_{i-1}}{T}$$

$$D_y = \sum_i^{N_{coll}} \frac{(y_i - y_{i-M})^2}{t_i - t_{i-M}} \times \frac{t_i - t_{i-1}}{T}$$

$$D_z = \sum_i^{N_{coll}} \frac{(z_i - z_{i-M} - \hat{W}_z(t_i - t_{i-M}))^2}{t_i - t_{i-M}} \times \frac{t_i - t_{i-1}}{T}$$

Decorrelation distance M



- A well-simulated electron will explore configuration space ergodically, and eventually be uncorrelated with its past self.
- Measuring separation between an electron and itself far enough back in time is approximately the same as measuring separation between two different electrons.
- Thus MagBoltz makes a swarm prediction by simulating just one electron, for a lot of collisions (typically tens of millions).
- Cross sections must include energy dependent elastic, excitation, ionization, attachment, and have appropriate angular distributions.
- Simulations possible in generic configuration of E and B fields (which influence how the electron moves between collisions).

Other details.

The stochastic computer simulation of ion motion in a gas subjected to a constant electric field

H. R. SKULLERUD

Electron and Ion Physics Research Group, Physics Department, Norges Tekniske Høgskole, Trondheim, Norway

MS. received 24th July 1968

- *Magboltz also uses the Skulleruds Null Collision method for better speed.*

- Magboltz also has a comprehensive database of accurate cross sections.
  - *Periodically uploaded to LXCat, tuned by S. Biagi.*

Chemical Physics 398 (2012) 148–153

Contents lists available at ScienceDirect

Chemical Physics

journal homepage: www.elsevier.com/locate/chemphys

ELSEVIER

The LXCat project: Electron scattering cross sections and swarm parameters for low temperature plasma modeling

S. Pancheshnyi [a,b], S. Biagi [c], M.C. Bordage [a,b], G.J.M. Hagelaar [a,b], W.L. Morgan [d], A.V. Phelps [e], L.C. Pitchford [a,b,*]

[a] Université de Toulouse, UPS, INPT, LAPLACE (Laboratoire Plasma et Conversion d'Energie), 118 route de Narbonne, F-31062 Toulouse cedex 9, France
[b] CNRS, LAPLACE, F-31062 Toulouse, France
[c] University of Liverpool, Oliver Lodge Laboratory, Oxford St., Liverpool L69 7ZE, UK
[d] Kinema Research and Software, P.O. Box 1146, Monument, CO 80132, USA
[e] JILA, NIST and University of Colorado, Boulder, CO 80309-0440, USA

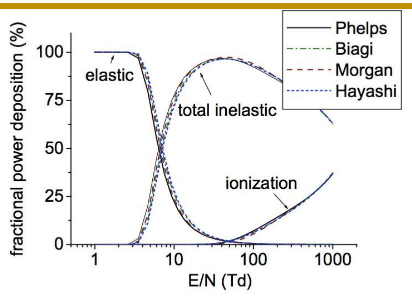


Fig. 3. Fractional power deposited into elastic, inelastic and ionization in argon vs *E/N* calculated using the different cross section sets as indicated.

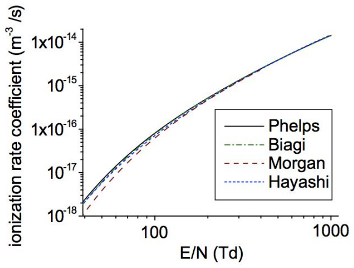


Fig. 4. Same as for Figs. 1 and 2, but showing the calculated ionization rate coefficients.

All the lines marked "Biagi" in LXCat are MagBoltz exported cross sections.

magboltz-10.6.f

MONTET Done Replace

```
      DZCOM=(VEZ-VGZ)*CONST11
C
C  CALCULATE  POSITIONS AT INSTANT BEFORE COLLISION
C    ALSO UPDATE DIFFUSION  AND ENERGY CALCULATIONS.
      T2=T*T
      IF(T.GE.TMAX1) TMAX1=T
      TDASH=0.0D0
      A=AP*T
      B=BP*T2
      SUME2=SUME2+T*(E1+A/2.0D0+B/3.0D0)
      CONST7=CONST9*DSQRT(E1)
      A=T*CONST7
      CX1=DCX1*CONST7
      CY1=DCY1*CONST7
      X=X+DCX1*A
      Y=Y+DCY1*A
      Z=Z+DCZ1*A+T2*F1
      ST=ST+T
      IT=DINT(T+1.0D0)
      IT=DMIN0(IT,N300)
      TIME(IT)=TIME(IT)+1.0D0
C ENERGY SPECTRUM FOR 0 KELVIN FRAME
      SPEC(IE)=SPEC(IE)+1.0D0
      WZ=Z/ST
      SUMVX=SUMVX+CX1*CX1*T2
      SUMVY=SUMVY+CY1*CY1*T2
      IF(ID.EQ.0) GO TO 121
      KDUM=0
      DO 120 JDUM=1,NCORST
      ST2=ST2+T
      NCOLDM=NCOL+KDUM
      IF(NCOLDM.GT.NCOLM) NCOLDM=NCOLDM-NCOLM
      SDIF=ST-STO(NCOLDM)
      SUMXX=SUMXX+((X-XST(NCOLDM))**2)*T/SDIF
      SUMYY=SUMYY+((Y-YST(NCOLDM))**2)*T/SDIF
      IF(J1.LT.3) GO TO 120
      ST1=ST1+T
      SUMZZ=SUMZZ+((Z-ZST(NCOLDM)-WZ*SDIF)**2)*T/SDIF
  120 KDUM=KDUM+NCORLN
  121 XST(NCOL)=X
      YST(NCOL)=Y
      ZST(NCOL)=Z
      STO(NCOL)=ST
      IF(NCOL.GE.NCOLM) THEN
       ID=ID+1
       XID=DFLOAT(ID)
       NCOL=0
      ENDIF
C ---------------------------------------------------------------------
C     DETERMINATION OF REAL COLLISION TYPE
C ---------------------------------------------------------------------
      R3=drand48(RDUM)
C FIND LOCATION WITHIN 4 UNITS IN COLLISION ARRAY
      CALL SORTT(KGAS,I,R3,IE)
  140 I=I+1
      IF(CF(KGAS,IE,I).LT.R3) GO TO 140
      S1=RGAS(KGAS,I)
      EI=EIN(KGAS,I)
      IF(IPN(KGAS,I).LE.0) GO TO 666
C  USE FLAT DISTRIBUTION OF  ELECTRON ENERGY BETWEEN E-EION AND 0.0 EV
C  SAME AS IN BOLTZMANN
      R9=drand48(RDUM)
      EXTRA=R9*(E0K-EI)
      EI=EXTRA+EI
C IF AUGER OR FLUORESCENCE ADD EXTRA IONISATION COLLISIONS
      IEXTRA=IEXTRA+NC0(KGAS,I)
C
```

The Down-side of MagBoltz.

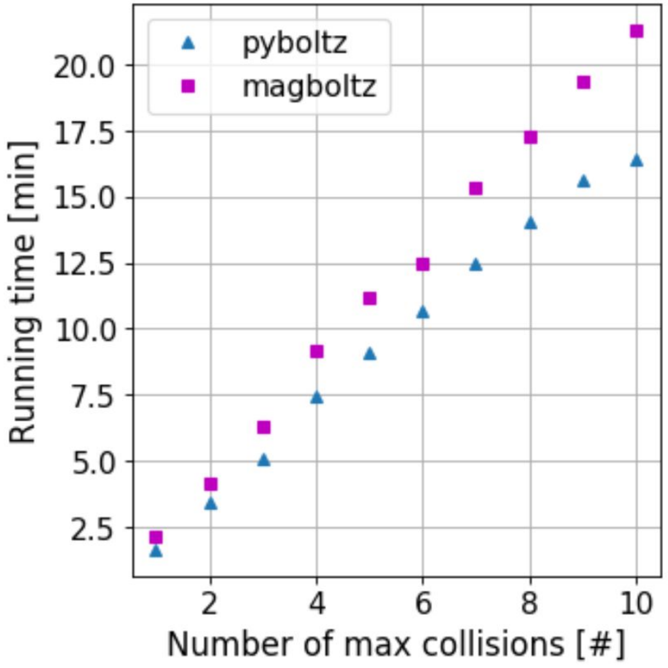
~82000 lines of FORTRAN

Everything hard coded and cryptically named..

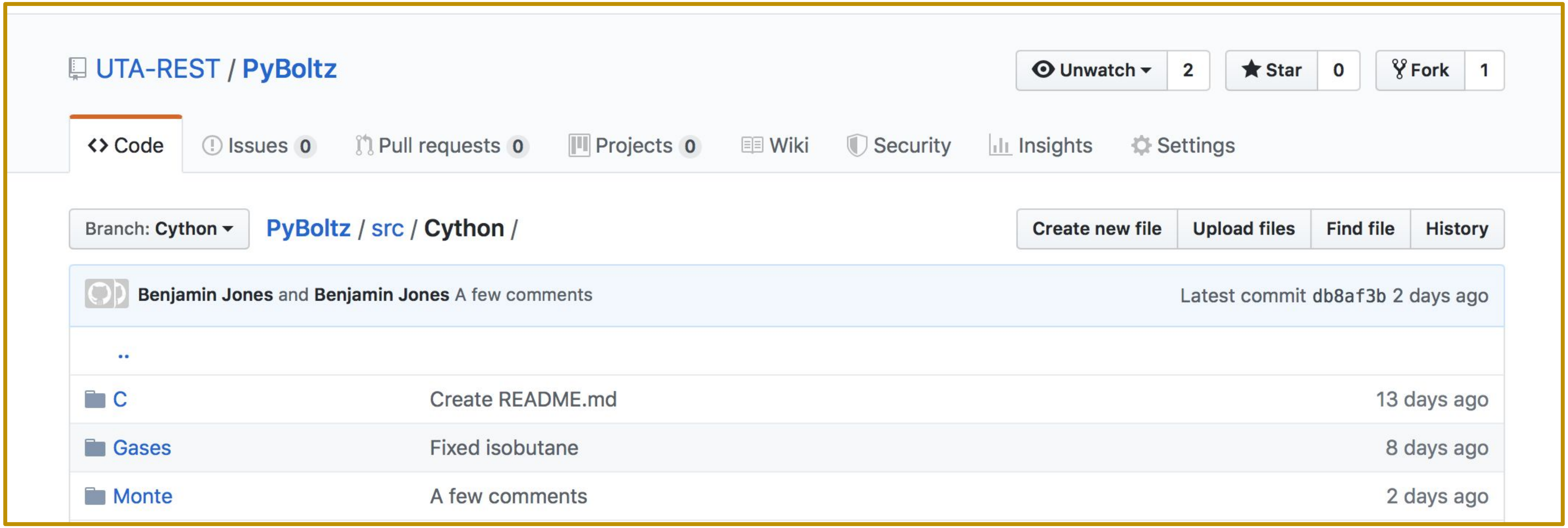
- It is easy enough to run. However, it is almost impossible to modify / extend / interface / understand.
- Garfield++ made a wrapper for Magboltz, yet it only improved the ease of running it. it was still hard to do any modifications.

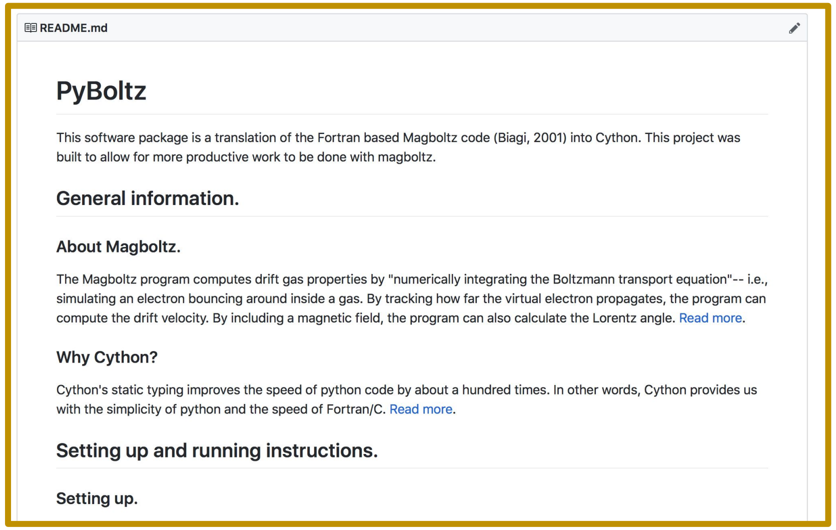
PyBoltz: The new MagBoltz.

- Full refactorization of entire MagBoltz code base into Python.
  - Modular structure, English variable names.
  - On par performance with Magboltz. (Cython)




```
# We run collisions in NumSamples batches,
# evenly distributed between its MaxNumberOfCollisions collisions.
CollisionsPerSample = <long long> (Object.MaxNumberOfCollisions / Object.NumSamples)
for iSample in range(int(Object.NumSamples)):
    for iCollision in range(int(CollisionsPerSample)):
        while True:

            # Sample random time to next collision. T is global total time.
            RandomNum = random_uniform(RandomSeed)

            # This is the formula from Skullerud
            T = -log(RandomNum) / Object.MaxCollisionFreqTotal + TDash
            TDash = T

            # Apply acceleration.
            #
            #     VBefore = VBefore + a t
            #      EAfter = 1/2 m VAfter^2
            #             = 1/2 m(VAfterX^2 + VAfterY^2 + VAfterZ^2)
            #             = 1/2 m((VBeforeZ + at)^2 + VBeforeX^2 + VBeforeY^2)
            #             = EBefore + (dir_z)(AP + BP * T) * T
            #
            #  w/      AP = m VBefore a
            #          BP = 1/2 m a^2
            #
            # So here, F2 = sqrt(m / 2) EField e
            #          BP = 1/2 m EField^2 e^2
            #

            AP = DirCosineZ1 * F2 * sqrt(EBefore)
            EAfter = EBefore + (AP + BP * T) * T
            VelocityRatio = sqrt(EBefore / EAfter)
```





PyBoltz has extensive documentation on both sphinx and github.

https://github.com/UTA-REST/PyBoltz

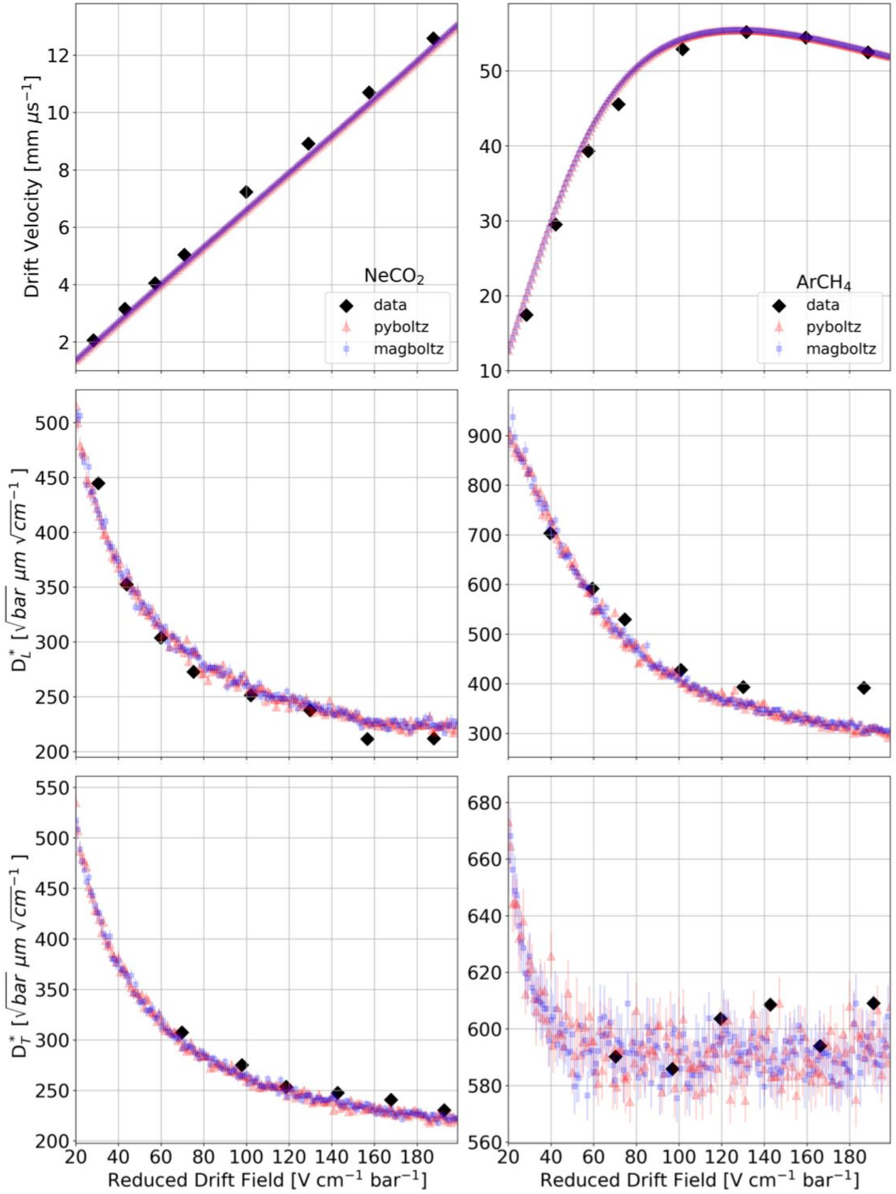


Figure 3: Left: the drift velocity, longitudinal and transverse diffusion of 91%Ne 9%CO_2. Right:The drift velocity, longitudinal and transverse diffusion of 90%Argon 10%CH_4. All calculations are at standard temperature and pressure. Data sets were taken from Ref [28].

Comparisons.

- Ran PyBoltz and Magboltz with different configurations, and confirmed both results to data.
- More tests were ran through the development phase.

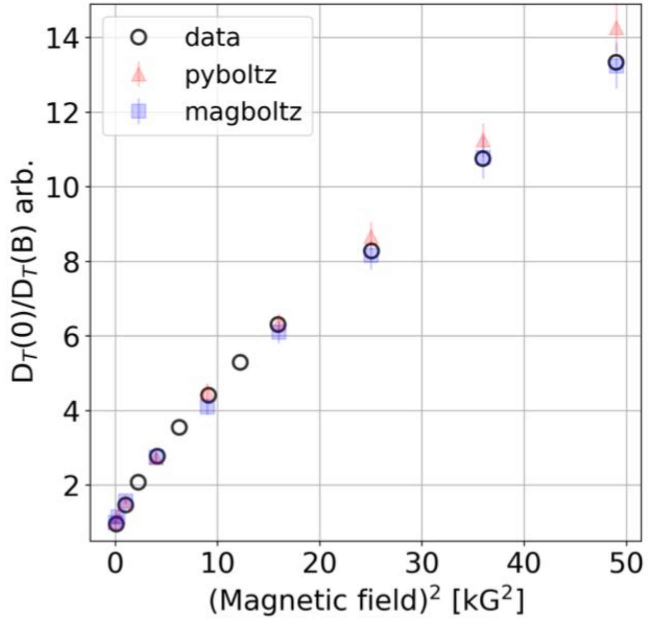


Figure 4: Validation of B-field suppression of transverse diffusion in Ar-CH_4 mixtures, **MagBoltz** and **PyBoltz** compared to data of Ref. [29]

Projects/modifications using PyBoltz.

- Our first extension is incorporation of modified effective range theory for cross section tuning.

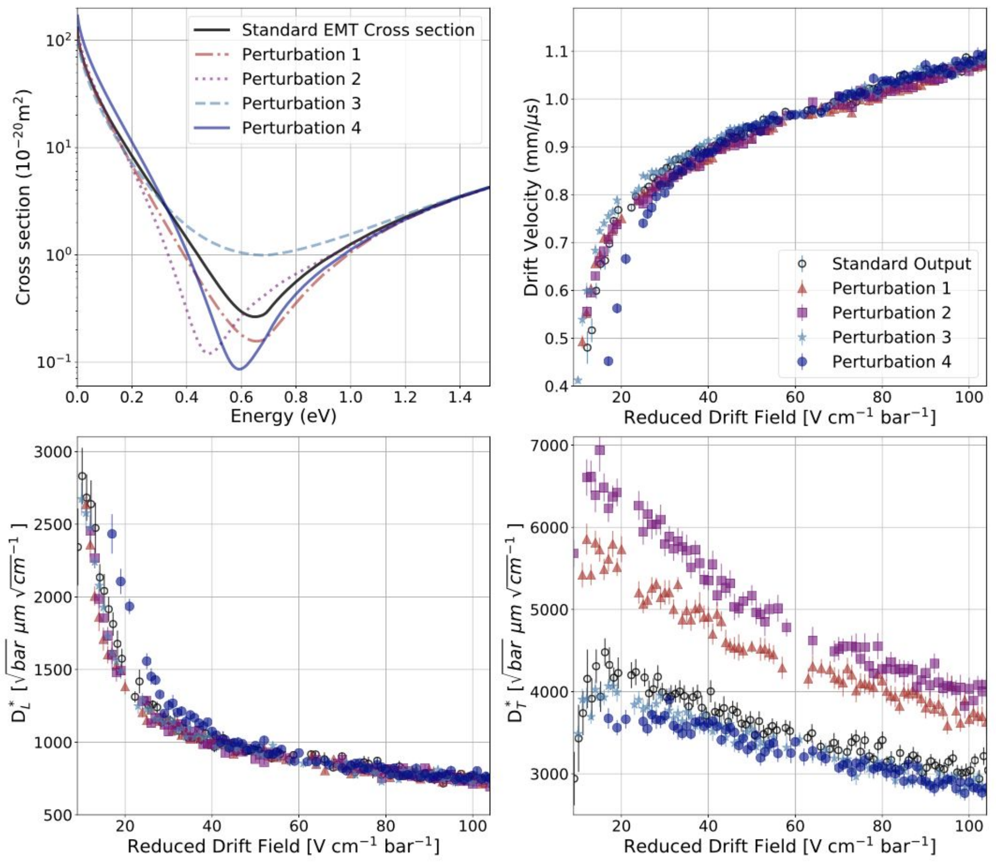


Figure 5: Top left: Default momentum transfer cross section in PyBoltz alongside different parametrizations consistent with the MERT formalism. Top right, bottom left, and bottom right display the effect of changing cross sections on drift velocity, longitudinal, and transverse diffusion respectively.

Method of cross section parametrization consistent with angular momentum conservation in quantum mechanics.

Spherical wave phase shifts

$$\tan(\eta_0) = -Ak[1+\frac{4\alpha}{3a_0}k^2 ln(3a_0)] - \frac{\pi\alpha}{3a_0}k^2 + Dk^3 + Fk^4$$

$$\tan(\eta_1) = \frac{\pi}{15a_0}\alpha k^2[1-\left(\frac{\varepsilon}{\varepsilon_1}\right)^{\frac{1}{2}}]$$

$$\tan(\eta_l) = \pi\alpha k^2/[(2l+3)(2l+1)(2l-1)a_0]$$

Momentum transfer and total cross sections

$$\sigma_m = \frac{4\pi a_0^2}{k^2}\sum_{l=0}^{\infty}(l+1)\sin^2(\eta_l - \eta_{l+1}),$$

$$\sigma_t = \frac{4\pi a_0^2}{k^2}\sum_{l=0}^{\infty}(2l+1)\sin^2(\eta_l).$$

Argon- and Xenon- Plus Anything

- Rapid-fire exploration of diffusion reducing gas mixtures based on argon and xenon.
- Jumping off point for further UTA gas mixture studies targeted at DUNE MPD and NEXT.

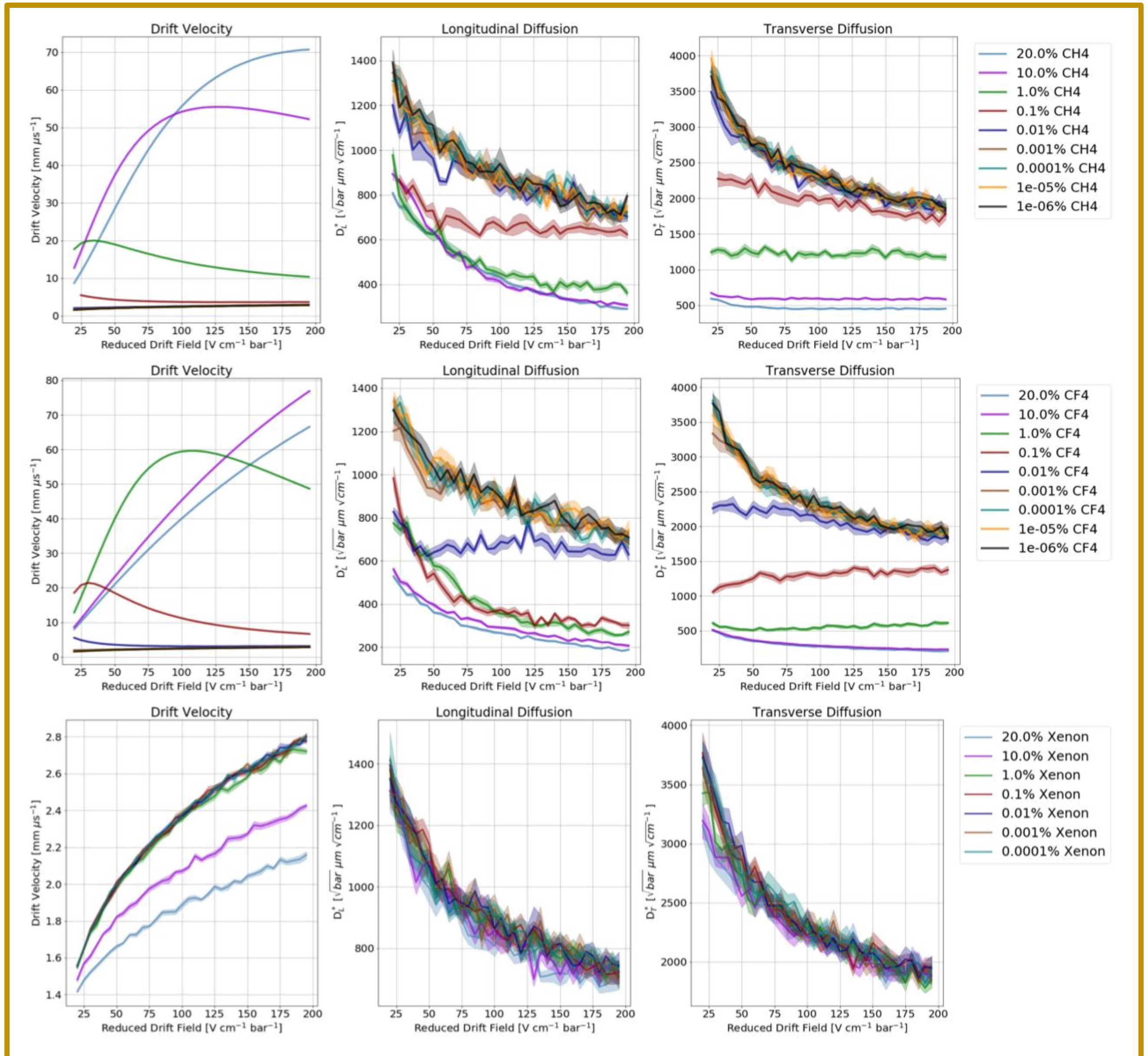
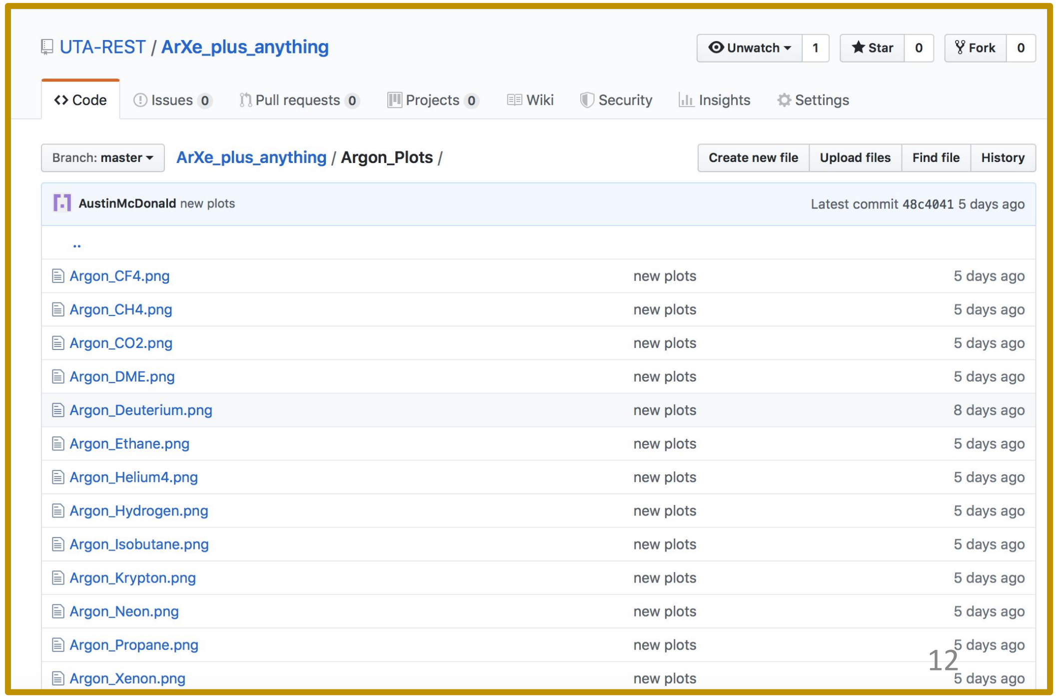
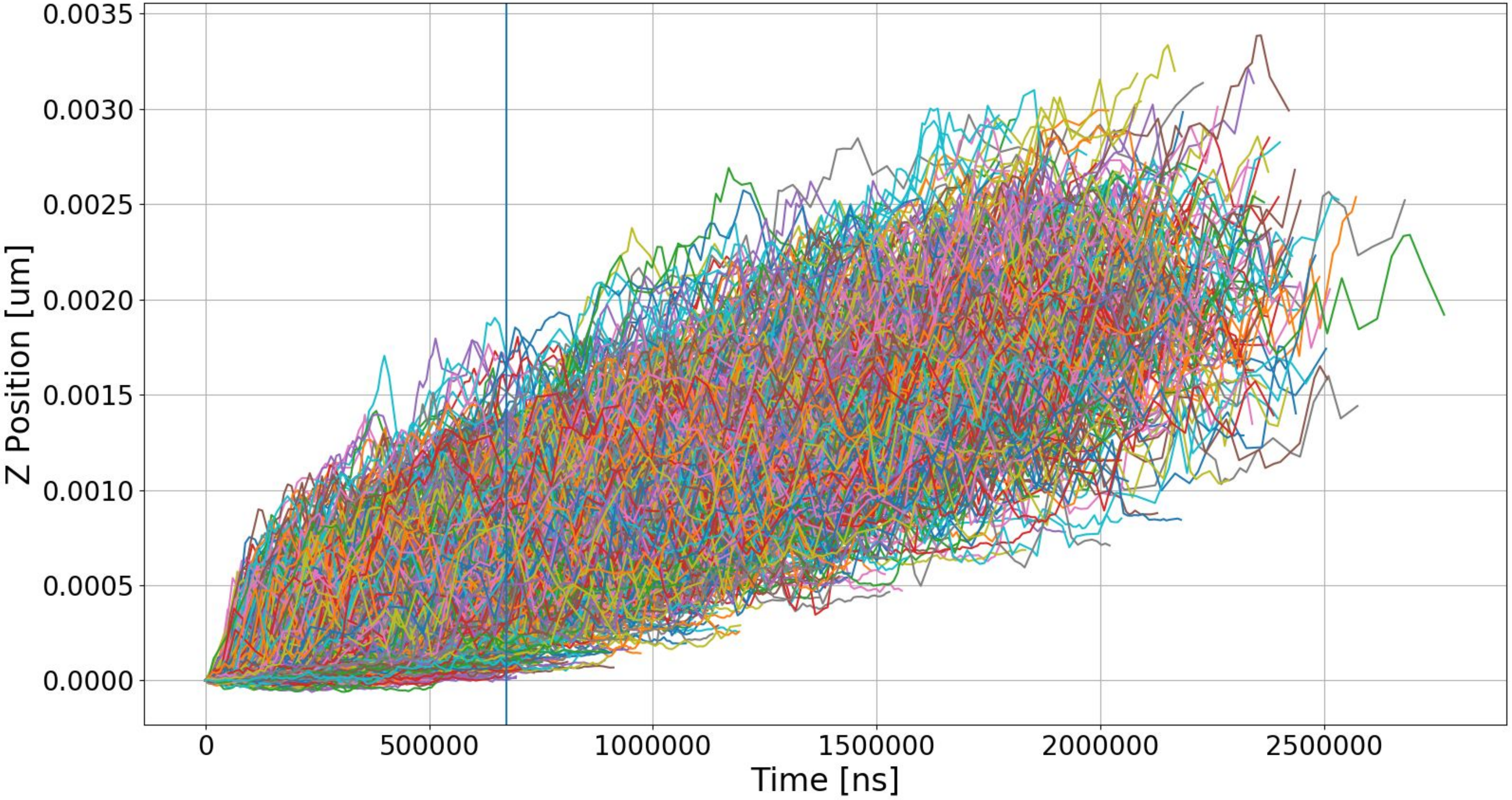


Figure 6: Example plots from the "Argon Plus Anything" project using PyBoltz. This example uses argon with various concentrations of CH_4, CF_4 and xenon. We show here drift velocity, longitudinal and transverse diffusion coefficients.



Multi-electron simulation.

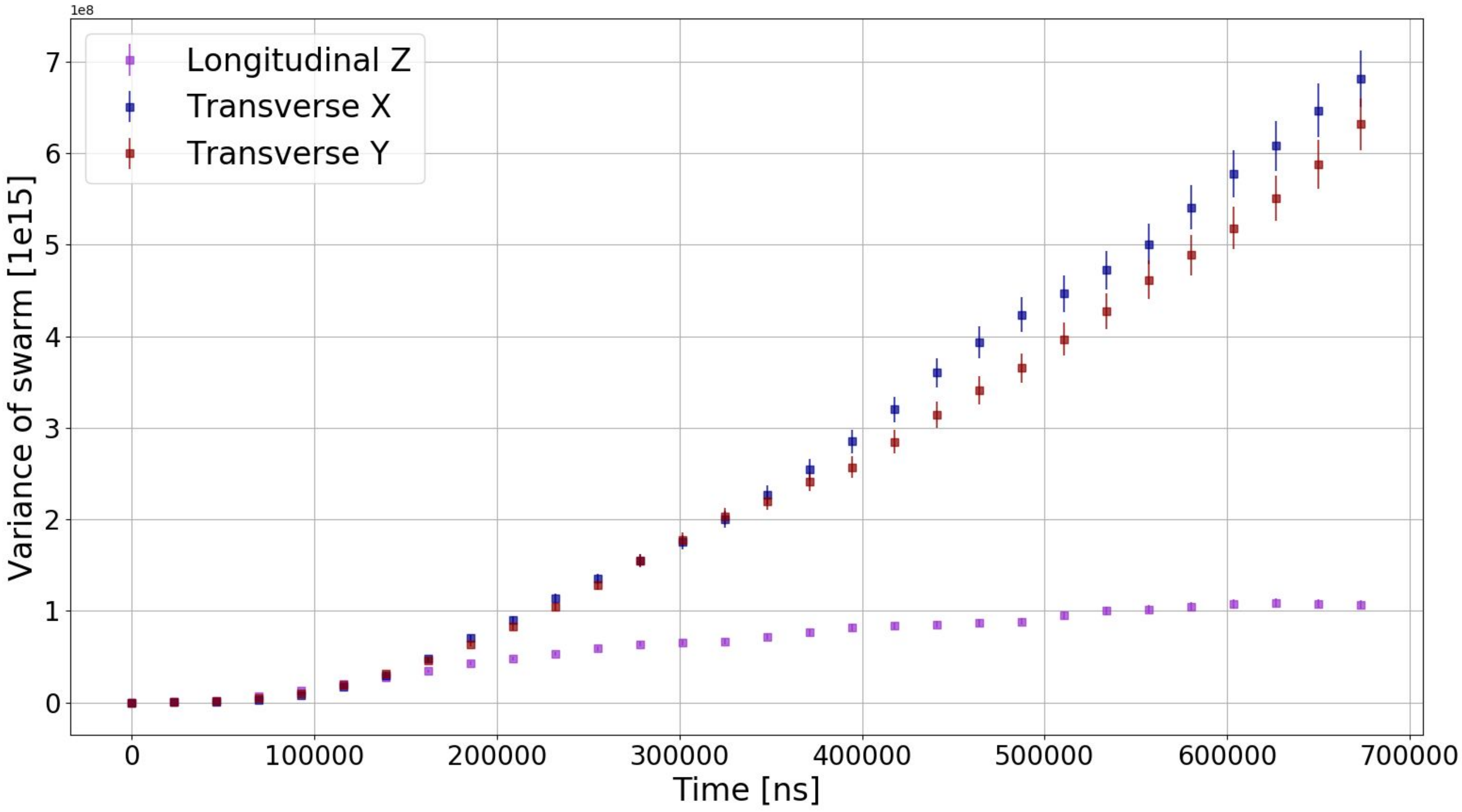


Z position of a swarm of 1000 electrons

- Using a Gpu to simulate multiple electrons. Gpus can theoretically run a huge amount of threads simultaneously.

- This method is a more direct swarm simulation than the de-correlation method, and should be equivalent in ideal cases.

Cont.



Variance of the different positions over time

- By calculating the slope of the variance over time line, the program calculates the diffusion coefficients.

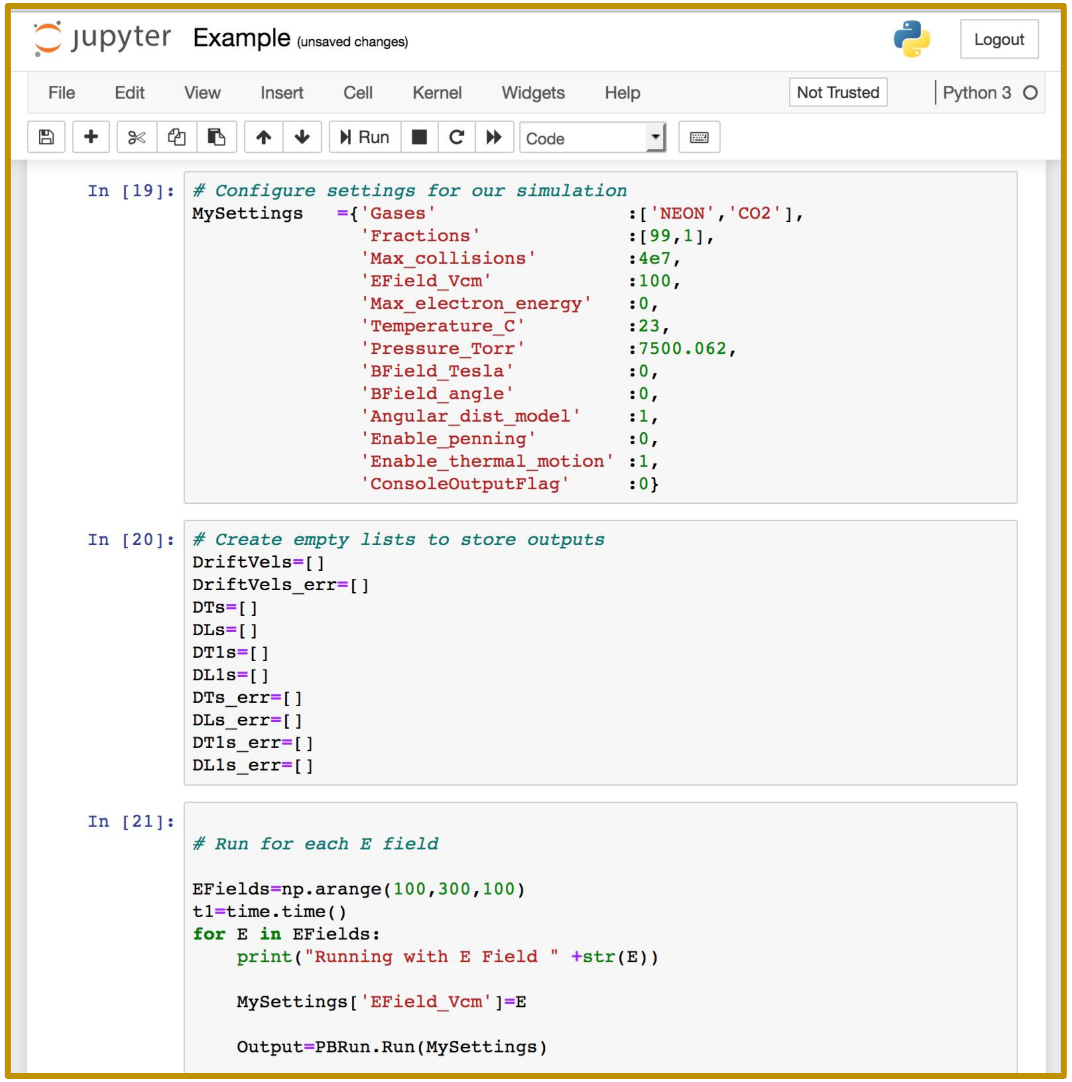
Future plans.

- Improving the modularity of the current gas functions, and their respective cross sections. That is by implementing a universal gas functions that only needs the cross sections as an input.

- Improving the results of the Xenon gas, through testing different cross sections, and different angular distributions.

Getting started

- Easy-to-use examples based on Jupyter notebooks and raw python in the Examples folder.
- We will support new users to get started – the sooner the better!
- Also open to ideas, thoughts, questions about interesting gas simulation studies. Please reach out if you have ideas!



Thanks!