This is a bounded one-dimensional kinetic particle-in-cell code with Monte-Carlo collisions and secondary electrons that is capable of accounting for any type of species, as well as any number of different types. Users can supply functions or constants that return the collision cross-sections and secondary electron yields. Sample Argon and atomic Hydrogen functions are supplied. Code features include:

* Elastic electron-atom scattering
* Electron-impact ionization
* Elastic ion-atom scattering
* Ion-atom charge-exchange collisions
* Electron-surface reflection
* Electron-surface back-scattering
* Electron-impact secondary electrons
* Ion-impact secondary electrons
* Optional oblique magnetic field
* Two bounded walls with user defined constant or time-dependent voltages
* Optional injection source with user defined injection options
* Optional steady state boundary where an electron-ion pair is put into the simulation domain every time an electron arrives at a surface.
* Option to pick up a code run from where the previous one left off
* When upper particle limit is reached the population is reduced and particles are re-weighted. The user has the following options:
  + Select the number of velocity groups
  + Select the maximum number of particles allowed per group
  + Select the amount to decrease the population by
* User defines system and gas parameters via input files
* Built in diagnostics calculates graph data by taking averages over a period of time that is determined by the user. This graph data is printed to text files for use by data visualization applications. Diagnostics include:
  + Rho, (net charge density)
  + Phi, (electrostatic potential)
  + Electric field
  + Averaged species velocity through domain
  + Averaged particle velocity distributions at locations chosen by the user
  + Instantaneous quantities in time:
    - Total particle number of each species
    - Electrostatic potential at three locations chosen by the user
    - Charge density for each species at two locations specified by the user

Known Current Issues:

* The number of ionization events sometimes needs to be limited to a user defined value to avoid runaway ionization.

This code is still a work in progress. Future developments include:

* Have the user define everything in a single input file, via third party software. Currently several input files are used to define parameters and user-defined functions.
* The code which decides whether or not an electron collision occurs is still a work in progress due to the need to limit the number of ionization events. Currently an elastic collision occurs if a random number between zero and one is less than the elastic probability and likewise for ionization. In the case of ionization a further condition that the number of ionization events is less than a user defined limit is also enforced.
* Clean up code: some variables are re-declared in different classes.
* The current method that returns a particle velocity randomly taken off a Maxwellian distribution is a temporary place holder for more accurate techniques.
* Option to watch real time graphics data.
* User defined graph quantities.
* More collision types.
* The ability to handle relativistic velocities.
* Looking into reworking as C code for efficiency.