^{4}He-DFT BCN-TLS user Manual. Training school version

Ernesto García Alfonso, François Coppens

June 11, 2024

Contents

1 Prerequisites

1.1 General

- Linux (prefered) or Windows with Visual Studio or other environment for Fortran development
- Gnuplot (for plots)
- Intel MKL/FFT
- Intel Fortran compiler (iFort).
  - Available for Linux or Windows architecture, no longer for Mac OS `https://www.intel.com/content/www/us/en/developer/tools/oneapi/hpc-toolkit-download.html?operatingsystem=linux&linux-install-type=online`

1.2 Obtaining and Compiling the codes

Obtain the codes from the COSY training school repository, `https://elibeamlines-my.sharepoint.com/personal/maria_krikunova_eli-beams_eu/_layouts/15/onedrive.aspx?ga=1&id=%2Fpersonal%2Fmaria%5Fkrikunova%5Feli%2Dbeams%5Feu%2FDocuments%2FOrganisatorisches%2FHe%20nanodroplets%20summer%20school%202024%2FFrom%20lectures%20%E2%80%93%20supporting%20materials%20for%20trainees%2FNadine%20Halberstadt`

1. Go to Nadine Halberstadt directory
2. Download all the content from the Ernesto Garcia Alfonso repository to a dedicated HeTDDFT directory in your computer
3. Check you have all the folders listed below
   - Ar_He500_dynamics
   - Ar_He500_statics
   - Code_dynamics
   - Code_statics
   - plotting_programs
   - Pure_He500_statics
   - DFT.pdf (this file)

Note: The programs are maintained in the GitHub repository `https://github.com/4He-DFT-BCN-TLS/He_DFT_2024`

2 ^{4}HeDFT statics simulations

2.1 First calculation: A pure ^{4}He$_N$ droplet

The easiest calculation to do is a pure ^{4}He$_N$ droplet. A typical droplet size used to study desorption dynamics and collisions is about $N = 10^3$ atoms. It is large enough to exhibit the desired physical properties and at the same time small enough to keep it computationally feasible. We start here with $N = 500$ for the sake of time.

The program will setup a simulation grid in cartesian coordinates. The helium density will be determined at each point of the grid, which should be large enough to contain the droplet. The mesh size should be ~ 0.4 Å.

2.2 Preparing the calculation directory

It is good practice to create a separate directory for each calculation, to keep all the input and output files together. First go back to your ^{4}He-TDDFT home directory. Next, go to Code_statics directory and type in a terminal

```
$ make
```

After this, you should have an executable called "4hedft" executable.

Returning to ^{4}He-TDDFT home directory, create a folder called "static" that will contain all the static calculations. Then create a subdirectory "pure-4he500" to host your first calculation:

```
$ mkdir static
$ cd static
$ mkdir pure−4he500
$ cd pure−4he500
```

In the code directory Code_statics/ in your ^{4}He-TDDFT home directory there is a folder named "input_files" which contains the input parameters for the calculation (pure-4he1000.input) and a script to submit the calculation to the computer cluster (pure-4he1000.sbatch). Copy these files into your calculation directory and make a "symbolic link" to the program:

```
$ cp ../../Code_statics/input_files/pure−4he1000.input .
$ cp ../../Code_statics/input_files/pure−4he1000.sbatch .
$ ln −s −f ../../Code_statics/4hedft .
```

Next, examine the input variables in the "pure-4he1000.input" file.

2.3 The minimal set of input parameters

Here follows a description of a minimal set of parameters that need to be setup for a successful calculation. There are more parameters than there are in this file. The full set of parameters that can be controlled will be given and explained later.

Table 1: **General Information**

title	the title of the calculation
nthread	number of OpenMP threads (CPU's) to use
mode	**0-** continue a previous calculation while reading all the input parameters from the density. An input density should be provided. This allows to stop and restart a calculation **1-** build, from scratch, a density for a pure ^{4}He$_N$ droplet **2-** build, from scratch, a density for a ^{4}He$_N$ droplet and a quantum impurity **3-** build, from scratch, a density for a ^{4}He$_N$ droplet and a classical impurity **4-** continue a calculation but ignore the impurity's position in the input density file and use the position given in the input file/impurity-wave-function. Can be used to find e.g. • the ground state of a droplet with an impurity (classical only), starting from a pure droplet, or • the ground state of a displaced impurity (classical/quantum) under a constraint, starting from an unconstrained ground state **5-** build a density for a ^{4}He$_N$ droplet with a classical impurity starting from a previously converged pure ^{4}He$_N$ droplet density. **6-** build a density for a ^{4}He$_N$ droplet with n classical impurities (n $\geq$ 2) starting from a previously converged pure ^{4}He$_N$ droplet density. Modes {2-6} are not relevant for a pure droplet but only mentioned here for completeness

Table 2: **Grid**

{x,y,z}max	the grid goes from -{xmax,ymax,zmax } to {xmax -dx, ymax -dy, zmax -dz } so the size in each dimension is $\sim$ 2 {xmax, ymax, zmax }. Typically $\approx$ 40 Å for 1000 He atoms
{n$_x$,n$_y$,n$_z$}	number of grid points in the {x,y,z} direction. Typically $\approx$ 200. The number needs to be a "magical number" dictated by the Fast Fourier Transform library. Ideally $n = 2^k$ or $n = 2^k + 2^l$ with $l < k$

Table 3: **Droplet properties**

n4	number of ^{4}He atoms. Typically n4= 10^3 (here, N=500)
$\{$x$_c$,y$_c$,z$_c\}$	droplet center-of-mass position. Usually $\{0,0,0\}$ unless the problem requires something else.
core4	the ^{4}He-functional to be used . This parameter is set to “OT” automatically if “lsolid” is set to “.true.” [1] • OT This is the default , no α_s term in the Orsay-Trento functional [2] • OTC Full Orsay-Trento functional [2], very CPU expensive • OP Orsay-Paris functional [3]
lsolid	switch to *.true./.false.* to enable/disable the use of the Solid Functional. When it’s enabled , “OT” is used regardless of the value of *core4* [4]
lexternalpotential	*.true./.false.* if we want to read or not an external file containing the potential Impurities-^{4}He$_N$
afermi	only important for densities starting from scratch. Its default value is set to 0.5 Å and it corresponds to the surface thickness of the initial density. Set to zero for a square density (surface thickness $\emptyset$)
limp	treat the impurity classically (=.false.) or quantum mechanically (=.true.)
lexternal	treat the classical impurity as an external field (=.true.)

Table 4: **Output-control**

pener	print an energy overview every *pener* $\in \mathbb{N}$ iterations in the standard output file
pdenpar	writes a partially converged density every *pdenpar* $\in \mathbb{N}$ interactions
filedenout	filename of the final converged density

Table 5: **Computational parameters**

niter	total number of iterations *niter* $\in \mathbb{N}$. When *niter* is reached, the program is stopped, even if the density is not converged
precie	precision of the convergence $0 <$*precie* $\in \mathbb{R} \ll 1$ is the size of the error in the total energy of the helium. The calculation stops when the error in the total energy becomes less or equal to this value. Typically $\sim 10^{-7}$ K
pafl	imaginary timestep size $\tau \in \mathbb{R}$. Usually $\tau \sim 0.05$
lpaflv	switch to allow for the imaginary time-step to change during the calculation, see next rows
nstepp	number of imaginary time-step changes
/	
1000 0.0001	use an imaginary time-step of 0.0001 for the first 1000 iterations
2000 0.0005	an imaginary time-step of 0.0005 for the next 2000 iterations
3000 0.0010	an imaginary time-step of 0.001 from iteration 3000 on

2.3.1 Execution and output files

The executable can be run in two ways. It can be run on the current machine:

```
$ ./4hedft < pure-4he1000.input > pure-4he1000.res 2> pure-4he1000.err &
```

"&" means that the program will be executed in the background so we keep the terminal free for other use.

It can also be run in batch mode on a computer cluster. The batch manager, here SLURM, will schedule the job and run it. The script is provided with the code and you already copied it in Section 2.2: *pure-4he1000.sbatch*. To submit the job :

```
$ sbatch pure-4he1000.sbatch
```

You can view the status of your job by invoking

```
$ squeue -u <user-ID>
```

In both cases, standard output will be written to file *pure-4he1000.res* and the errors to *pure-4he1000.err*. The program will write output files that are listed in the following table

Table 6: **Output files**

DFT4He3d.namelist.read	a complete list of all the input variables, including the ones that were not explicitly given in the input file
den-{x,y,z}.dat	1-dimensional density profile in the {x,y,z} direction at the time of the last written partial density
den.out (or "filedenout")	the converged output density file
den0-{x,y,z}.dat	1-dimensional density profile in the {x,y,z}-direction at the start of the calculation
partial.density{1,2,...}	partially converged density at {1,2,...}×pdenpar
density.{iteration#}.out	partially converged density after an iteration number of imaginary time step
pure-4he1000.error	file containing the messages to *stderr* generated by the program
pure-4he1000.res	file containing the messages to *stdout* generated by the program

To check if the calculation is running okay, check if the error in the total energy of the system is decreasing. This is written every *pener* iterations. in the *pure-4he1000.res* file, as

```
ITERATIVE PROCEDURE
Total Energy (He).......... -5400.344271 K +/- -1.7363E-06 K
```

As long as the error is steadily decreasing, the calculation is running fine. The calculation will terminate whenever this value becomes smaller than or equal to *precie* or if the number of iterations reaches *niter*.

- **Note:** If the imaginary time step is too large, the calculation might "explode" : restart it again with a smaller time step from the beginning or from a previous intermediate density file where the error was still decreasing.

- **Note:** If storage space is a problem keep only the last few intermediate density files for a possible restart and remove the first ones.

2.4 A droplet with a classical impurity X-^{4}He$_N$

Building the ground-state density for a droplet with a classical impurity can be done in two ways. Building the whole system from scratch, or using an existing droplet and adding the impurity. The easiest method is the first one, while the fastest method is the second one; when adding the impurity to a converged pure droplet, it only needs to adapt locally, as long as the initial location of the impurity is not chosen too poorly.

In either case, the following parameters need to be changed and added

```
lexternal = .false. -> lexternal = .true.
```

and the following six new parameters in the input file need to be set

Table 7: **Additional parameters for a classical impurity**

mimpur	impurity mass in unified atomic mass units (u). E.g, potassium is ~39 u
rimpur	radius of the impurity. Usually the distance (X-He) at the bottom of the X-^{4}He pair potential well is a good choice, about $3-4$ Å
{x,y,z}imp	location of the center of mass of the impurity relative to the calculation grid
selec_gs	the ground state pair interaction potential to use. These can be looked-up in the file "V impur.f90" in the code repository
r_cutoff_gs	cut-off radius for the X-He interaction potential, usually set to several thousands of K
umax_gs	cut-off value for the X-He interaction potential (see r_cutoff_gs)

As a consequence, there are also six new output files containing information about the interaction between the impurity and the droplet

Table 8: **Output files**

uext-{x,y,z}.dat	1-dimensional cuts of the external potential between the impurity and the helium at the start of the calculation
uext-{xy,yz,xz}.dat	same as *uext-{x,y,z}.dat* , but two-dimensional

2.4.1 Building a X-^{4}He$_N$ system from scratch

To build a X-^{4}He$_N$ system from scratch, one additional input parameter needs to be changed:

```
mode=1 -> mode=3,5,6
```

input file called "cx-4he1000-from scratch.input" is already prepared for this in the *input_files* directory. Now setup your calculation directory as explained in section 3.1.2 and start the calculation by executing

```
$ cd your He-TDDFT directory
$ cd static
$ mkdir cx-4he1000-fs
$ cd cx-4he1000-fs
$ cp ../../Code_statics/input_files/cx-4he1000-from_scratch.input .
$ cp ../../Code_statics/input_files/cx-4he1000-from_scratch.sbatch .
$ ln -s ../../Code_statics/4hedft 4hedft
$ sbatch cx-4he1000-from_scratch.sbatch
```

Check the status of the job as before

```
$ squeue -u <cecam user ID>
```

and check the convergence in the output file in "*cx-4he1000-from_scratch.input*"

Another possibility is to run the executable (4hedft), on your computer, as :

```
$  ./4hedft  < ck-Ar4he500-from_scratch.input >
 ck-Ar4he500-from_scratch.res 2> ck-Ar4he500-from_scratch.err &
```

2.4.2 Building a X-^{4}He$_N$ system from a converged pure ^{4}He$_N$ droplet

To build the X-^{4}He$_N$ system starting from a converged pure ^{4}He$_N$ droplet, change mode parameter: (see Table [1])

mode=1 –> mode=4 or 5

and add one extra parameter to the *input-file*

Table 9: **Added parameters**

filedenin	filename of the converged input density of a pure droplet

Prepare your calculation directory as in the previous section, with one extra addition. You need to provide the file containing the converged density of a pure ^{4}He$_N$ droplet. This can be done in 3 ways:

1. copy the density file to the calculation directory
2. make a symbolic link from the actual density file to den.in in the calculation directory (this is the chosen option in the example below)
3. give the full path in the input file name

Be careful that the box size and number of points of the input density correspond to the box size and number of points in the input file.

The *input_files* directory contains an input file already prepared for this: '*'cx-4he1000-from_density.input*". Prepare the directory and start the calculation as before:

```
$ cd static
$ mkdir cx-4he1000-fd
$ cd cx-4he1000-fd
$ cp ../../code/4hedft/input_files/cx-4he1000-from_density.input .
$ cp ../../code/4hedft/input_files/cx-4he1000-from_density.sbatch .
$ ln -s ../../code/4hedft/4hedft 4hedft
$ ln -s ../pure-4he1000/den.out den.in
$ sbatch cx-4he1000-from_density.sbatch
```

Another possibility is to run the executable (4hedft), on your computer, as :

```
$  ./4hedft  < cx-Ar4he500-from_scratch.input >
cx-Ar4he500-from_scratch.res 2> cx-Ar4he500-from_scratch.err &
```

3 ^{4}He-TDDFT (dynamics)

From a practical point of view, the dynamics is simpler than the statics. It only requires an input density file and the setting of some initial parameters. It will not check whether the density is converged or not or if it is an eigenstate of the Hamiltonian. It will simply take the input data and propagate it in time, even if the inputs are silly or make no physical sense. As long as the errors keep withing bounds, it keeps on going. It assumes the user knows what he or she is doing. In the following two sections we will give one simple dynamical processus an example, that of an argon (Ar) atom colliding with a helium droplet containing another argon atom, to form a solvated argon dimer 5.

3.1 Ar-Ar@^{4}He$_{500}$ collision

We will now prepare and execute the dynamics simulation of a frontal (head-on) Ar atom (projectile) against a doped droplet Ar@^{4}He$_{500}$. The initial velocity of the Ar is set to 500 m/s (5 Å/ps). In order to prepare this dynamics you need two things:

- Ar_He500_dynamics directory
- a converged density of an argon-doped droplet (hosting an Ar atom inside)

Downloading and compiling the code is done exactly as before. For this it is assumed you are in your home-directory again. Execute the following series of commands

```
$ cd Code_dynamics
$ make
```

After this, you should have an executable called "*4hetddft-isotropic*" .

There is an example input file in the input files directory called *"ar-he500-head on.input"*. Some of the fields that are in the input file are omitted here because they are already explained in the static chapter, while others are included again because they have changed meaning.

Table 10: **New (or changed) input parameters**

mode	the different read modes to start or continue a calculation. **0-** start a dynamic calculation. You have to provide a file containing a density or wave function from a converged static calculation. **2-** continue a dynamic calculation. You have to provide a previously written file containing the wave-function of the droplet and an impurity in an isotropic electronic state **3-** same as **mode = 2** but using a file containing a wavefunction of a droplet and an impurity in an anisotropic electronic state. In this case the electronic state of the impurity will be read, but no longer propagated in time. You can use this mode to study the dynamics of an impurity that underwent a transition from an anisotropic electronic state. This case will not be discussed today.
selec	related to the selec_gs parameter in the statics. This field selects the He-X interaction potential. Can be looked up in V_impur.f90. E.g. for Ar this is "Ar_He"
r_cutoff	He-X interaction potential cutoff distance
umax	He-X interaction potential cutoff energy
v{x,y,z}imp	initial velocity of the impurity in ps/Å (1 ps/Å=100 m/s)
deltatps	the time step in picoseconds. A typical value is $5 \cdot 10^{-4}$
filerimp	name of the output file containing the impurity location as a function of time. Default name is rimp.out.#, where the # is an ID to distinguish between different runs. But you are free to name it as you want
filevimp	name of the output file containing the impurity velocity as a function of time. Default name is vimp.out.#
fileaimp	name of the output file containing the impurity acceleration as a function of time. Default name is aimp.out.#, where the # is an ID to distinguish between different runs. But you are free to name it as you want
pcurr	this parameter is related to pdenpar discussed before but now it prints a complex wave function instead of a helium density. The filenames of these files are density.#.dat, where the # is an integer ID.
txmean, tymean, tzmean	sets the location of the mean of the absorbtion potential (see Ref. 6,7 "Absorbing potential at the box boundaries"). A good rule of thumb is to use : txmean = xmax-2 ,tymean = ymax-2 ,tzmean = zmax-2 Å
txsurf, tysurf, tzsurf	It is defined as the distance from (txmean, tymean, tzmean) to the simulation box boundary. E.g : The absorption potential should act from [-tzsurf,tzsurf] around tzmean.
Λ_0	Damping absorption parameter, typically = 2, see [6]

To prepare our calculation directory and start the simulation execute the following commands

```
$ cd Ar_He500_dynamics
$ ln -s ../../Code_dynamics/4hetddft-isotropic
$ ln -s ../../Ar_He500_statics/static/
      pure-4he1000/den.out den.in
$ sbatch dynamic-sbatch
```

Another possibility is to run the executable (4hetddft-isotropic), on your computer, as :

```
$ ./4hetddft-isotropic < Ar2He500.input > Ar2He500.res 2> Ar2He500.input.err
```

The program will generate the following output files.

Table 11: **Output files**

density.#.dat	file containing the full 3-dimensional wave-function of the helium droplet and also the position and velocity of the atomic impurity, if present (otherwise these are set to 0). The file has a header containing information on the number of iterations, the used time-step, the current evolution time, etc, at the time of writing. This information can be extracted when needed. The "#" in the filename is an integer ID that labels the files. The time difference between two files is always deltatps×pcurr.
filerimp	output file containing the impurity position as a function of time. It is structured in 4 columns: time(ps), x(t), y(t), z(t). The coordinates are referred to the origin of the grid.
filevimp	output file containing the impurity velocity as a function of time. It is structured in 4 columns: time(ps), v_x(t),v_y(t), v_z(t). The coordinates are referred to the origin of the grid.
fileaimp	output file containing the impurity acceleration as a function of time. It is structured in 4 columns: time(ps), a_x(t),a_y(t), a_z(t). The coordinates are referred to the origin of the grid
Energies	it is set as 6 columns : eimpu-impu (impurity-impurity energy), eimpu (impurities total energy), ekinx (impurity total kinetic energy), eHeX (He-impurity/ies interaction energy), etot (total energy)
hedenini-{x,y,z}	1-dimensional cuts along the box axis of the initial helium density.
*.err	file containing the run-time errors
*.res.#	file containing the results. # is an integer ID labeling the different runs of the program in case a calculation has been interrupted, either intentionally or unintentionally
timec.dat	file containing the time coordinates
uext.dat	Fourier transform of the helium-impurity interaction potential
uimp-{x,y,z}	helium-impurity interaction potential

3.2 Continuing a previous dynamic calculation

Table 12: **New input parameters**

icurr	ID of the current input wave-function. If file containing the droplet's wave function (and λ-vector if applicable) that you are going to use to continue is called "density.123.dat", then icurr=123
iter0	current number of iterations. This number is included in the header of the wave-function file.
time0	current evolution time This number is also included in the wave-function file

4 Annex

4.1 He$_N$ droplet with several classical impurities X$_M$

In cases where we deal with multiples impurities, we can use the following program : `https://github.com/4He-DFT-BCN-TLS/He_DFT_2024/tree/main/External_potential`. You can store it in a different folder from the main DFT programs.

This folder called here My_directory could contain the following subfolders:

- My_directory which contains the following folders:

1. External_Potential: (X$_M$-He$_N$) potential interaction
2. cx-4he1000-fd : where the DFT calculations.

Table 13: **Input and Output files to build the X$_M$-He$_N$ Potential**

imp.input	It contains: the He-X$_M$ interaction potentialat each point of the droplet grid. Impurities positions, r_cutoff_gs_k and umax_gs_k
potential_multi_impurity.input	Number of impurities, number of points of the grid, {xmax,ymax,zmax} for the He box.
potential_multi_impurity.out	Output file which contains the interaction potential of all X$_M$-He$_N$ for every grid point

You must define the following variable in the static input file (*ck-4he1000-from_scratch.input*) :

```
expotential ='potential_multi_impurity.out'
```

As above, you could copy the full interaction potential in the working directory, make a symbolic link or give the full path of the potential filename in the input file.

4.1.1 Input file containing the impurity positions

An additional input file called ”imp.input” is required, in order to specify the impurity positions. We name it as **“imp.input”**. It is going to contain the positions in x,y and z for each impurity.

```
rimp(1,1) = x1,    rimp(1,2) =y1,    rimp(1,3) = z1
rimp(2,1) = x2,    rimp(2,2) =y2,    rimp(2,3) = z2
.
.
.
rimp(n,1) = xn,    rimp(n,2) =yn,    rimp(n,3) = zn
```

4.2 Internal Units of the Programs

We give here some details on the units used in the calculations. In the helium functional definition the energy unit is Kelvin and the length unit is Å . In the dynamics the time scale is most often ps, and hence velocities appear in Å/ps, (1 Å/ps = 100 m/s), although the program works in other units, see below. Some ingredients such as pair potentials have to be written in these units before being used. The energy scale of the absorption and emission spectra is cm^{-1}. Let us recall some equivalences and constant values taken from the **NIST** table of constants (`http://physics.nist.gov/constants`):

- Recall that: 1Å= 10^{-10}m ; 1fm= 10^{-15}m; 1ps= 10^{-12}s
- Boltzmann constant: $8.6173303 \cdot 10^{-5}$ eV/K; hence 1 eV = 11604.522 K
- 1 cm^{-1} = 1.43877736 K
- Atomic mass unit: $m_u c^2$ = 931.4940954 MeV = $1.080954 \cdot 10^{13}$ K
- $\hbar = 6.582119514 \cdot 10^{-16}$ eV s =7.638235 K ps
- c=299792458 m/s
- $\hbar$c = 197.3269788 MeV fm = $2.289885 \cdot 10^{7}$ K Å
- He mass : $4.002 \times 931.49 = 3727.8$ MeV

When solving the effective Schrödinger equation for the superfluid, the $\hbar$ constant has been incorporated into the definition of “time”. This means that the equation actually solved is

$$i \frac{\partial \Psi}{\partial (t/\hbar)} = \mathrm{H}\Psi \tag{1}$$

For the sake of consistency, this way of handling time has also been adopted for the dynamic evolution of the impurity. As a consequence, what internally is called “velocity” of the impurity (v_{inter}) for instance, is actually $\frac{d\mathbf{r}}{d(t/\hbar)}$ instead of $\frac{d\mathbf{r}}{dt}$. One may recover the “physical” impurity velocity v_{phys} by dividing the internal velocity v_{int} by $\hbar$, namely $v_{phys} = v_{int}/\hbar$.

Similarly, in order to obtain the kinetic energy of the impurity I in K, as any other energy printed and/or stored during the dynamics, one has to keep in mind that

$$\frac{1}{2}m_I v_{phys}^2 = \frac{1}{2}\frac{m_I}{\hbar^2}v_{inter}^2 \rightarrow \frac{1}{2}\frac{m_I c^2}{(\hbar c)^2}v_{inter}^2 \tag{2}$$

Writing the impurity mass in units of m$_u$, m$_I$ = M$_I$ m$_u$ and taking for M$_I$ the value in atomic mass units as given in the atomic mass list of **NIST** (which takes into account the natural isotopic composition of the impurity), one obtains the kinetic energy of the impurity in K as

$$E_{kin,I} = \frac{1}{2}M_I \frac{m_u c^2}{(\hbar c)^2}v_{int}^2 \rightarrow \frac{1}{2}M_I \times 0.02061484\ v_{int}^2(\mathrm{K}) \tag{3}$$

since $\frac{m_u c^2}{(\hbar c)^2} = 0.02061484\ \mathrm{K}^{-1}\ \text{Å}^{-2}$. M$_I$ values for the impurities used in some recent applications are e.g. M$_{Rb}$ = 85.4678; M$_{Cs}$ = 132.9055; M$_{Ar}$ = 39.948 amu.

The kinetic energy of the impurity in K is usually evaluated from its velocity in hundreds m/s, which gives

$$E_{kin,I}(\mathrm{K}) = \frac{1}{2}M_I m_u c^2 (\frac{v}{c})^2 \rightarrow 0.6013618 M_I (v_{\mathring{A}/ps})^2 \tag{4}$$

For Cs at 200 m/s, v^2 = 4, M$_{Cs}$ = 132.9055 and this yields E$_{kin,Cs}$ ~ 319.3 K

4.3 Plotting Programs

```
sudo apt install gnuplot  (Linux)
sudo port install gnuplot (MacOS)
```

References

[1] F. Ancilotto, M. Barranco, F. Coppens, J. Elorante, N. Halberstadt, A. Hernando, D. Mateo, and M. Pi, *International Reviews in Physical Chemistry* **36**, 4.621 (2017).

[2] F. Dalfovo, A. Lastri, L. Pricaupenko, S. Stringari, and J. Treiner, *Physical Review B* **52**, 1193 (1995).

[3] J. Dupont-Roc, M. Himbert, N. Pavloff, and J. Treiner, *Journal of Low Temperature Physics* **81**, 1/2 (1990).

[4] F. Ancilotto, M. Barranco, F. Caupin, R. Mayol, and M. Pi, *Physical Review B* **72**, 214522 (2005).

[5] E. García-Alfonso, M. Barranco, D. A. Bonhommeau, N. Halberstadt, M. Pi, and F. Calvo, *The Journal of Chemical Physics* **157**, 014106 (2022).

[6] M. Barranco, F. Coppens, N. Halberstadt, A. Hernando, A. Leal, D. Mateo, R. Mayol, and M. Pi, “Zero temperature DFT and TDDFT for ^{4}He: A short guide for practitioners”, Unpublished, 2017.

[7] https://github.com/4He-DFT-BCN-TLS/DFT-Guide/blob/master/dft-guide.pdf.