**Context**

This report is a supporting document of the MATLAB program developed by Sam Borkent, based on the paper *“Numerical modelling of the plasma plume propagation and oxidation during pulsed laser deposition of complex oxide thin films”*, by Tom Wijnand, et al. (2020). The program was developed to fulfil the requirements of a 5 EC capita selecta, commissioned by Prof. Mark Huijben, of the Inorganic Materials Science (IMS) group of the University of Twente.

**Background**

Advances in modern technology are often driven by engineering materials with novel properties. One way to engineer the properties of a material is to reduce its dimensionality, as this constrains the propagation of waves and particles within the material to a certain preferential direction and alters its physical and statistical properties. For example, the dimensionality of a 3D bulk material can be reduced to a 2D thin film [?] or nanosheet [?], a 1D nanowire [?], or a 0D nanoparticle or quantum dot [?]. A well-studied and widely utilized technique to produce these low dimensional nanostructures is pulsed laser deposition (PLD). In PLD a high energy pulsed laser is focused on a target material in a vacuum chamber, the high energy density on the target surface causes the atoms in the target to ionize, resulting in the formation of an outward expanding plasma that propagates normal to the target surface. This plasma is then collected on a high temperature substrate surface, where it crystalizes into one of the aforementioned nanostructures. PLD enables epitaxial growth of a large variety of materials on a wide range of substrate materials. The most common materials that are being studied with PLD are complex metal oxides. This category of materials has been shown to exhibit exotic properties ranging from ferroelectricity [?] to superconductivity [?].

Another important specification of PLD is that is allows for stoichiometric transfer from target to substrate. This means that the composition of the target material is maintained during growth. To achieve this stoichiometric transfer a low pressure background gas is commonly introduced in the vacuum chamber. The primary functionality of this background gas is threefold. Firstly, if the kinetic energy of a particle arriving at the substrate surface is too high, it will undergo an elastic collision with the surface and bounce off. The background gas increases the number of collisions between particles which reduces the total kinetic energy of the plasma, so particles will stick to the substrate surface on arrival. Secondly, if oxygen gas is used, collisions with the background gas will result in oxidation of reactive species in the plasma, altering the oxidation state in which the plasma arrives at the substrate. Thirdly, the background gas reduces desorption of particles from the substrate, which can prevent oxygen depletion at the surface. In the case of oxygen gas, the background gas can also increase the oxidation of the substrate surface and potentially fill oxygen vacancies. These effects combined makes the choice of the background gas and background pressure some of the most important PLD growth parameters that can be tuned to produce the desired materials.

PLD can achieve high quality results quickly compared to other high-end methods, and is particularly suitable for development of complex oxide material, so increasing the scale of the technology is also being investigated. The aim is to produce homogeneous structures on large area wafers, so PLD can be applied for industrial chip and sensor fabrication. One major physical difference between large area and small scale PLD is that on a large wafer the entirety of the plasma plume is collected, while a small area substrate only receives the centre of the plasma. The size of industrial wafers is much larger than the physical width of the expanding plasma, so in large area growth the plasma is scanned over the wafer surface to ensure uniform coverage. The composition and density can vary spatially within the plasma, so collecting the entire plasma instead of only the centre will affect the material growth.

Modelling the propagation of the plasma particles and the interaction between the plasma and background gas could give insight into the spatial variation of the particle density and composition in the expanding plasma, which can be used to approximate the chemical composition and the kinetic energy of particles arriving at the substrate. This could provide a helpful tool during target selection and PLD growth parameter optimization.

**Model description**

First, the number of ablated particles are calculated, which are then angularly distributed. Then the initial velocity distribution is calculated per atom type in the target, and the plasma particles are placed at the target surface and assigned their respective velocities. The number of background gas particles are calculated and distributed evenly in space. Then the particle position is updated every timestep based on the particles velocity. For each timestep a collision calculation is performed between the plasma particles and the background gas.

*Plasma particles*

The total number of ablated particles per laser pulse is given by:

Here is the number of particles per unit cell, is the volume of the ablation spot given by the laser spot area times the ablation depth. The laser spot is assumed to be square in this model. There is an inverted relation between the height and width of the laser spot and the spatial expansion of the plasma, so a wider spot in one direction would result in a more narrow expanding plasma in that direction and vice versa, while in this model the plasma is assumed to be axially symmetric to the target surface normal. The ablation depth is taken constant at for the results shown, but for most accurate results should be determined per target. is the volume of the unit cell of the target material. If multiple unit cells are present in the target, is the average of the volume of the unit cells present in the target, weighted by the ratio between species in the target. So, for accurate modelling of targets consisting of multiple species the ratio between species should be known. is the measured density of the target, and is the single-crystal density of the materials in the target based on a perfect tiling of unit cells. If multiple species are present in the target is taken as a weighted average based on the ratio between species, similarly to .

As modelling every single particle individually (~1016 in total) is not feasible with current computers, particles are collected together in computational bins with other particles of similar type, location, velocity, and number of collisions. The spatial expansion of the plasma plume is modelled by dividing space into a number of angular bins ranging from 0° (normal to target surface) to 90° (parallel to target surface), see figure 1. An important approximation in this model is that particles only move forward within their initially assigned angular bin, thus all collisions are head-on, and no exchange between angular bins is possible. This approximation is a great oversimplification of reality, nonetheless the model is able to achieve results closely matching experiment. This could be explained by that in reality on average a similar number of particles exit and enter each angular bin each timestep. Also, collisions under a leftward or rightward angle will cancel out, resulting in a mean forward motion. So, a 1D problem is solved for each angular bin, which can be combined to obtain a 2D expansion model.

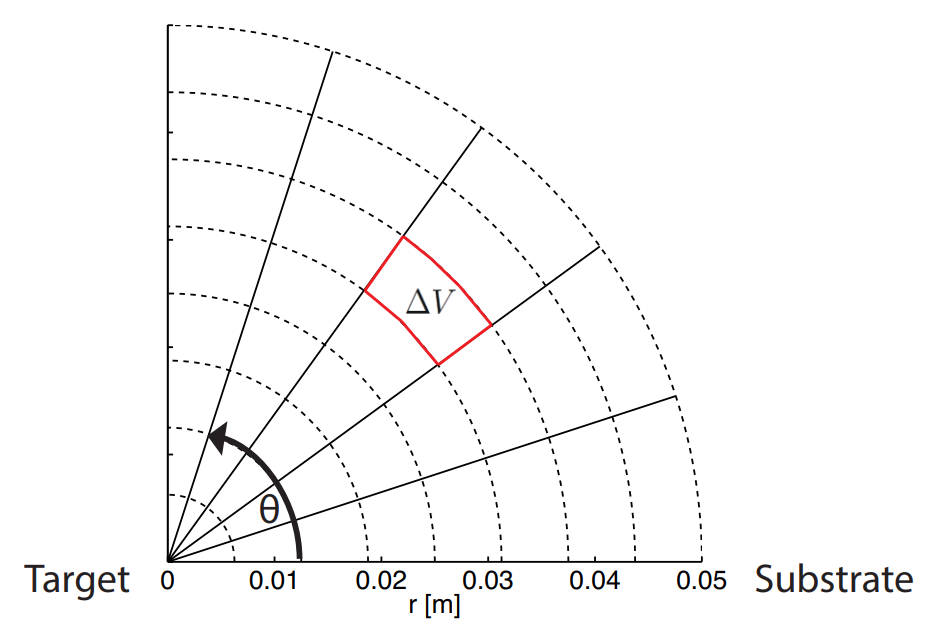


Figure 1 - Schematic of the polar grid which represent the spatial computational bins of the numerical model, where the horizontal axis show the distance between target and substrate.

The model does not include the intricate physics of the target ablation. Instead, it starts at after ablation, where the angular particle distribution is assumed to be approximately spherical [?]. This angular particle distribution is given by:

Here is the normalization factor, and is a fitting parameter used to match the plasma shape to experiment. A higher results in a more forward oriented plasma, which is predicted to correspond to a higher laser energy, although this is not investigated further here. Results shown use , which was determined by Wijnand from measurements of the expansion of a TiO2 plasma.

To calculate the initial velocities of particles after ablation an energy balance equation is solved:

Where is the ratio of energy absorbed by the target. Now we can take which is the energy of a single laser pulse. The absorbed energy can be expressed as:

Here is the heat going into the target. For ceramic targets the heat dissipation is generally small, so it can be neglected. could be approximated using the specific heat equation, although this requires knowledge of the specific heat capacity of the target and the amount of mass that is being heated. The activation energy can be expanded as follows:

Here is the number of unit cells in the ablation spot volume . A part of the energy goes into breaking apart the unit cell, and the remainder of the energy goes into the atoms in the unit cell. is the binding energy of the ablated unit cell. The sum is over all atoms in the unit cell, where part of the energy goes into kinetic energy and the other part goes into exciting electrons in the atom to higher energy levels. It is difficult to determine what the excitation energy per atom is, so is assumed to be small. Each atom in the unit cell receives an equal portion of the absorbed energy, so we can write the kinetic energy per atom of type as:

Which gives an expression for the initial average (RMS) velocity per atomic species using :

The plasma particles will initially follow distribution in velocity. In Wijnand’s model the distribution was assumed to be Gaussian with a fixed distribution width. This width was determined experimentally based on the expansion of titanium atoms. There are two problems with this approach. Firstly, if all particles are energized by the laser, no negative particle velocities are allowed. A Gaussian distribution spans negative to positive infinity, so this is not the proper distribution to use. Secondly, based on other diffusion processes a inversely proportional relation is expected between the distribution width and the particle mass, with a wider distribution for a lower mass and vice versa. So, a Maxwell-Boltzmann distribution was chosen instead with general form:

Where is a normalizing constant, is the mass of the ablated particle , the Boltzmann constant, and the thermodynamic temperature. The temperature can be related to the mean kinetic energy per particle using:

Resulting in the final expression for the distribution:

Note that for a target mixture with multiple different unit cells that contain the same atom it is possible to get one atom with multiple distinct initial velocity profiles, caused by the difference in binding energy between the unit cells. For example, consider a target that is composed of a mixture of unit cell () with a low binding energy, and unit cell () with a high binding energy that both contain atom . will allow more energy to be transferred to the atoms than , as less energy is needed to overcome the binding energy of as compared to . So, atoms of type originating from will have a higher average initial velocity than atoms originating from . This enables spatial separation of multiple particle fronts of atoms of the same type, which does not occur for targets consisting of a single unit cell.

Now we have an expression for the number of plasma particle of type at initial radial position and at time per velocity bin and angular bin :

Here is the number of collisions, which is initially zero, and is the number of atoms of type per unit cell.

*Background gas particles*

In the vacuum chamber a temperature gradient is present in the background gas from the high temperature substrate to the room temperature target. This sophistication is not included in the model currently, although it would be possible to implement by taking a different temperature for each radial bin. For the results presented here the initial temperature of the background gas is assumed to be constant throughout space with . Using a constant temperature, the particle density of the background gas can be approximated using the ideal gas law:

Here is the background gas pressure, and the Boltzmann constant. The initial average kinetic energy of the background gas can be related to the temperature via the equipartition theorem:

Where is the mass of the molecules in the background gas. Common background gasses are O2 or Ar with an atomic mass of about and respectively. At a constant room temperature this gives an average initial velocity of the background gas particles of . This value is small compared to the initial velocity of the plasma particles (). Additionally, at constant temperature the background gas particles do not have a preferential propagation direction. For these two reasons the average initial background gas particle velocity is assumed to be zero.

As can be seen in figure 1, not all radial bins have the same volume. To obtain the initial number of background particles per computational bin, has to be multiplied by the bin volume :

This gives us the following expression for the spatial distribution of background particles at time :

*Collision categories*

As mentioned before, one of the main approximation of the model is that all collisions are assumed to be head-on fully elastic collisions. Four distinct collision categories can be identified: (1) collisions between the plasma and the background gas particles, (2) collisions between particles of the same type, (3) collisions between different metals in the plasma, and (4) collisions between metals and oxygen particles in the plasma. Collisions of category 1 are the main contribution to the plasma dynamics, and are the only collisions included in this model. Category 2 collision can be safely neglected as a head-on elastic collision of two particles of the same mass will simply exchange their velocities, which does not result in a net change in particle density. Category 3 and 4 collisions are not included due to time constraints and computational restrictions, but could be implemented in the future.

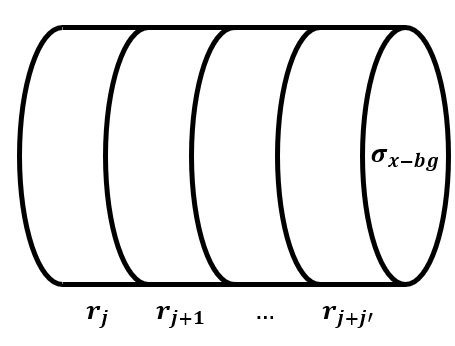
If two metal plasma particles are of similar mass, their initial average velocities will be close to each other, increasing the number of category 3 collisions. If the masses of the two metals are significantly different, their initial average velocities will be spaced apart, and consequently they will separate in space, reducing category 3 collisions. However, if a significant amount of low mass metal atoms gets slowed down due to collisions with the background gas, higher mass metal particles might still collide with the low mass metal particles, which would again increase the significance category 3 collisions. Category 4 collisions are more significant, as this will result in additional oxidation of plasma species.

*Number of collisions*

The number of type plasma particles of velocity that have collided with background gas particles of velocity at radial position , originating from position , at angle and time is:

Here is the number of remaining non-collided particles from the previous radial bin, and is the collision rate between the particles of type and the background gas given as:

Here is the particle density of the background gas with velocity at radial position . is a length of one radial bin, and is the scattering cross-section between the particle of type and a background gas particle, defined as the area of a circle with the sum of the two particle radii as radius. The product of and describes the *volume of the collision path* within a single radial bin, so multiplying this by the background density gives the mean number of background gas particles in the collision path. This can be viewed as an effective collision rate between the plasma and background gas particles. If two particles are moving in the same direction, along the same path, the probability of a high velocity particle colliding with a low velocity particle is higher than the probability of two particles of similar velocity colliding. To account for this the relative velocity term is added.



**Implementation**

*Data structure*

The data structure used to store all particles in the model has been completely reimplemented to improve performance, physical accuracy, and flexibility for using the model for any target composition. In Wijnand’s implementation there were two data structures: one for the plasma particles, and one for the background gas particles. The plasma structure was build up from MATLAB structs with fields for type of particle, number of collisions, radial position, and velocity. The background structure was similarly structured without the type of particle or number of collisions fields. There are a few problems with this method:

* The structs do not give insight into which values are stored in them, the only way to read values is to call them. This makes debugging difficult, and makes MATLAB’s convenient workspace viewer obsolete.
* The radial positions were called through the use of strings (‘X\_1’, ‘X\_2’, etc.). This requires the num2str function to be called millions of times, even though there is no real advantage of calling the radial positions in this way.
* The particles could have any number of velocities. Arrays with variable length are very slow, as every time the array gets resized its values have to be stored in a new place in memory. Also, every time the velocity is used it has to be rounded, so the particle position is a discreet value fitting the spatial and temporal resolutions. So, the number of velocities should be compatible with the length of the radial bins and the duration of one time step. ( travels one per , travels per , etc.) Any additional resolution in velocity is a waste of resources.
* Structs do not give a clear idea of how data is stored in memory. In modern computers the time it takes to access and write to memory has become the bottleneck in a lot of computations. If multiple values have to be called within one loop it can save a significant amount of time if these values are stored close together in memory. Using structs and using two different data structures for the plasma and background gas particles makes it difficult to guarantee this.
* The data structure used by Wijnand was specifically written for modelling the propagation of TiO2, and is not necessarily as suited for extension of the model to support any target material or composition.

For these reasons the decision was made to develop a new data structure. Instead of two separate data structures consisting of MATLAB structs, a single 4D matrix is used to contain all the particles. The dimensions are the particle type, number of collisions, velocity, and radial position in that order. Figure 2 shows a schematic overview of how the particle matrix is stored in memory. The main differences are as follows:

* A numerical matrix is considerably more efficient than multifield structs, as they have a fixed size, so they can be pre-allocated in memory, and a single value takes 4 bytes of memory instead of 8 bytes.
* The radial position is called with a numerical index instead of a string.
* Each particle can have a fixed number of pre-calculated velocities, compatible with the spatial and temporal resolutions.
* The background gas particles can be fully modelled just as the plasma particles, and can have every velocity, instead of only considering static or kinetic background gas with a single shared velocity. Also, the number of collisions of the background gas particles can be tracked.

**N**

**C**

**Par. 1**

**Par. N**

**N**

**C**

**Par. 1**

**Par. N**

**N**

**C**

**Par. 1**

**Par. N**

**N**

**C**

**Par. 1**

**Par. N**

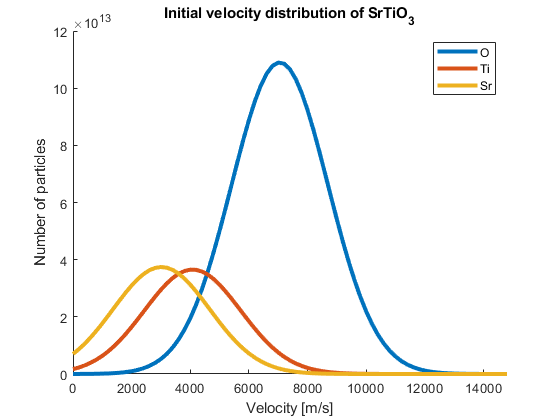
Figure 2 - Schematic view of how the data structure is stored in memory. U and C stand for uncollided and collided respectively.

* The plasma particles and background gas particles are stored in the same matrix. This saves time when performing collision calculations, as this requires to call the plasma and the background gas particles repeatedly, which goes quicker as their values are stored closer in memory.
* The structure is built with memory efficiency in mind, so values that are called often are stored closely together, and MATLAB’s efficient matrix operation can be used as frequently as possible.

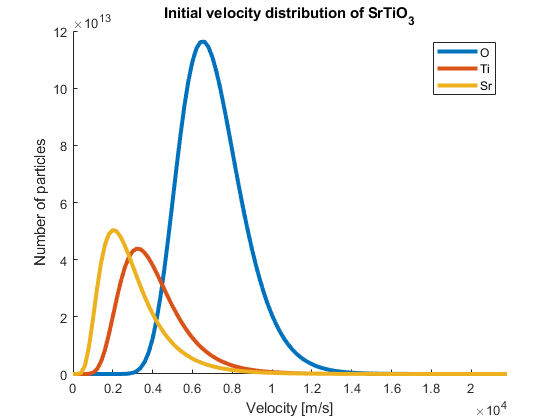
**Results**

*Initial velocity distribution*

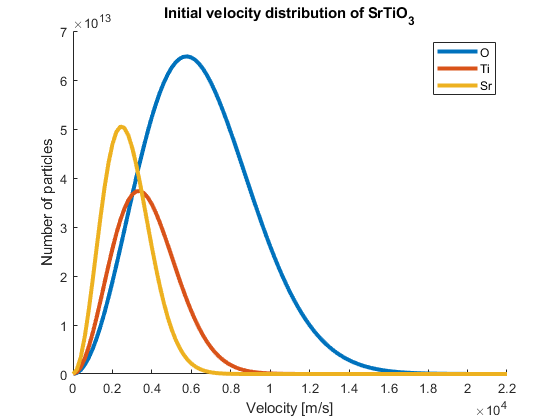
Normal distribution used in Wijnand’s paper. For heavy atoms with low initial mean velocity the distribution extends into negative values.



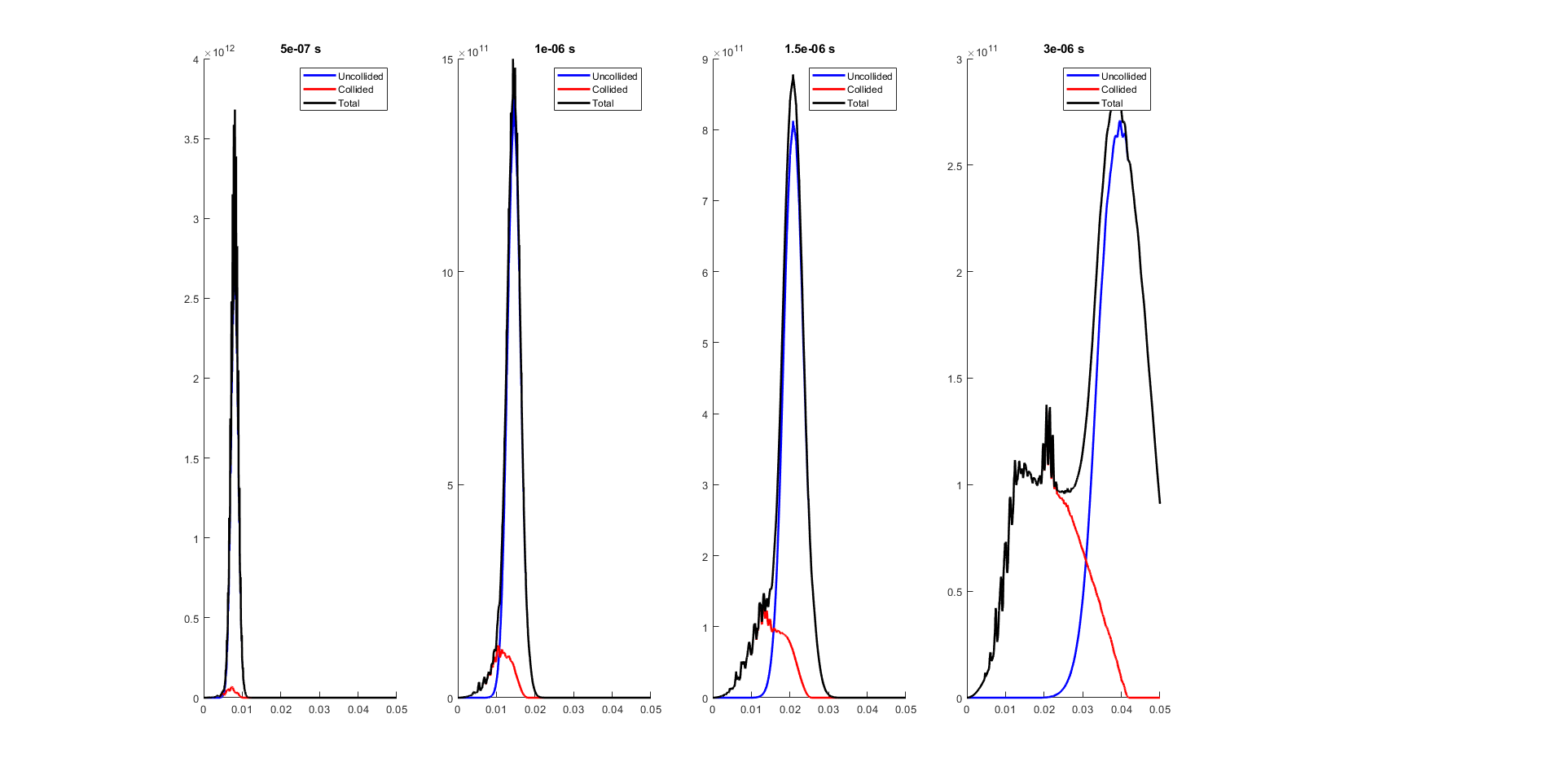
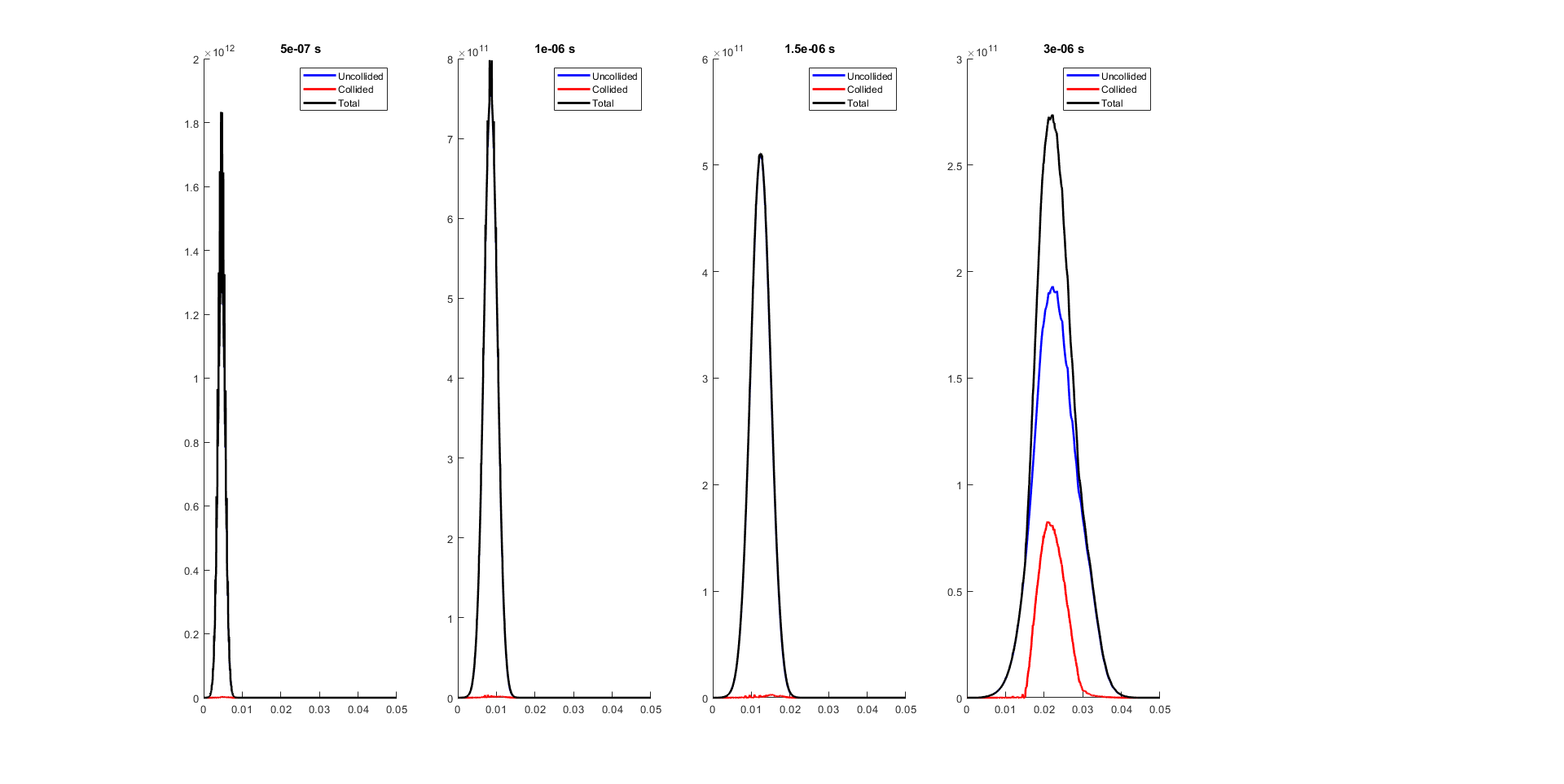
Log-normal distribution fixes this problem, as it can only have values in the range . Although, for higher mean velocities, the distribution approximates a normal distribution, and atoms of all masses will share the same velocity distribution width.

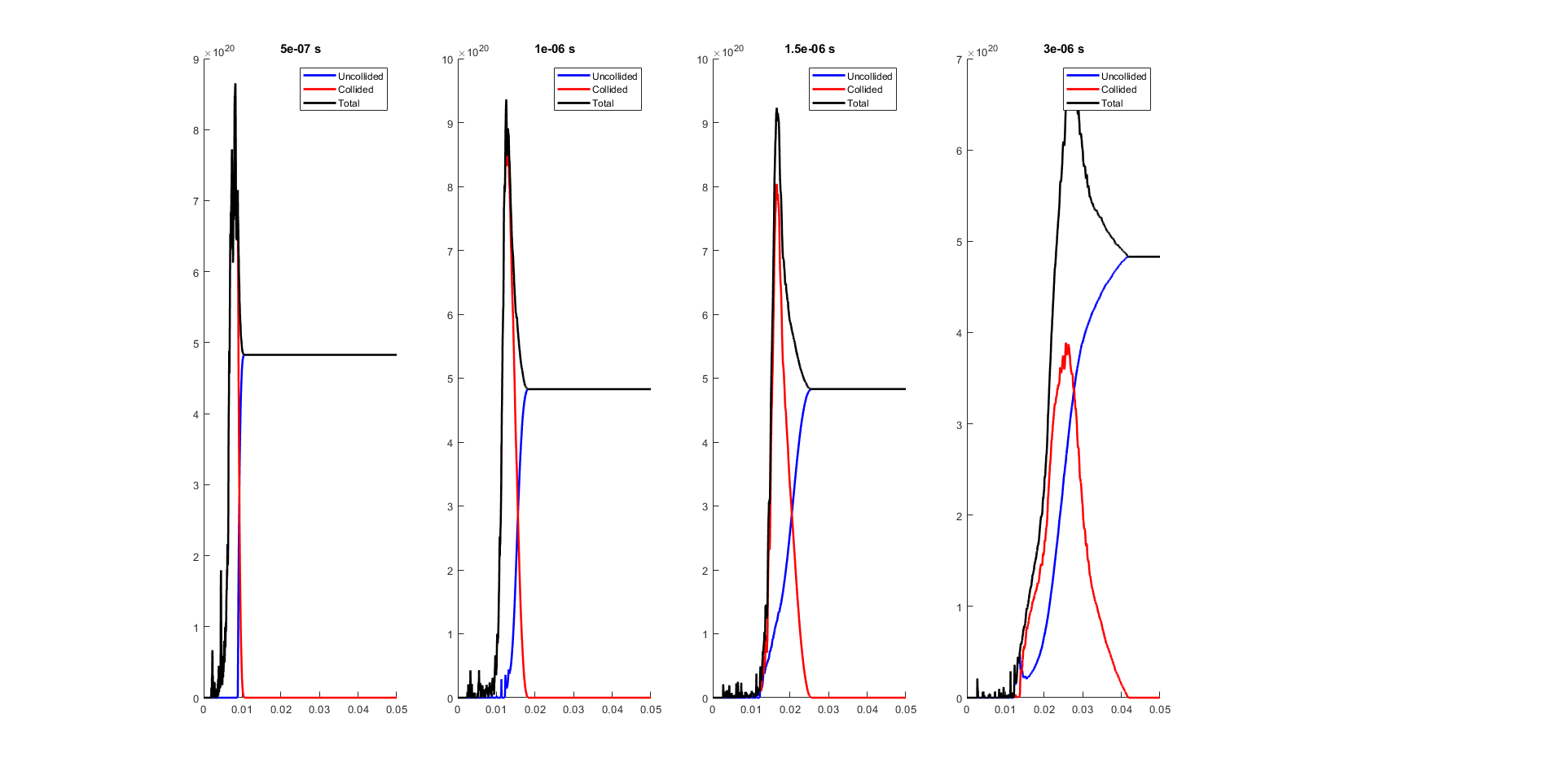


A Maxwell-Boltzmann distribution accounts for the problem of the negative extend of the normal distribution, and it gives a physical approximation of the distribution width inversely proportional to the atomic mass. However, the oxygen and metals are not clearly spaced apart as is the case with the normal distribution. This makes the assumption that the interaction between the different species in the plasma plume can be neglected questionable. Especially since the plasma particle density is orders of magnitudes higher than the background gas density.



The overlap between the curves of the different species can be seen as a ratio of plasma particles than could interact with each other through collisions if no collisions with the background gas take place. However, if light plasma particles collide with heavier background gas particles their velocity direction is inversed, and they will propagate towards the target, and towards the slower plasma particles. This could result in an increase in the number of inner-plasma collisions, although it is also possible for a backwards propagating plasma particle to collide again with a heavier background gas particles and regain a positive velocity.





**Conclusion**

*Approximations and assumptions*

* The laser ablation spot is square, resulting in plasma expansion that is axially symmetric to the target surface normal.
* There is no net exchange of particles between angular bins.
* All particles move along a straight line following their initial angle.
* All collision are head-on and fully elastic.
* Particles cannot move backwards.
* The background gas initially has a constant temperature.
* The background gas particles initially have no preferential propagation direction, resulting in a net zero velocity.
* Collisions only occur between plasma particles and the background gas particles.

*Changes and improvements compared to Wijnand’s model*

* Improved angular distribution, removed term from
* Changed data structure from a struct to a multi-dimensional matrix that stores all particles, including background gas particles.
* Fully model background gas particle kinematics, so with the same number of velocity bins as plasma particles
* Supports targets of any composition as long as the target density and the material properties of the species in the target are known. Targets fabricated from a ratio of different material are supported as well, if the ratio between the species is known.
* The model is developed with conservation of particles in mind. After initialization no particles are destroyed and no additional particles are created.
* Fixed some mistakes.
* There were multiple discrepancies between the code and what was described in the paper. The formulas shown in this report accurately reflect the formulas used in the new code.

*Features missing compared to Wijnand’s model*

* 2D plume expansion plots
* Oxidation of plasma species

*Work in progress*

* Code clean-up and factorizing the code into functions.
* Testing and finalizing the functionality of multi-component targets.
* Implementation of oxidation of plasma species.

*Possible improvements*

* Introducing a temperature gradient in the background gas based on the substrate temperature.
* Finding an approximation model for the excitation energy applied to target atoms during ablation.
* Implementing collisions between plasma particles, most importantly between oxygen and metals in the plasma, so oxidation of species from oxygen originating from the target can be accurately modelled.
* Allowing particles to have a negative velocity, so they can move in two directions, which is essential for accurate modelling of propagation of low mass particles like Li or O atoms, as their propagation direction will be invert after collision with a heavier particle. This can be implemented either by extending the velocity axis for negative values, or by adding an additional dimension to the data structure which holds the propagating direction of the particles. This poses a challenge due to how the particle propagation is currently implemented. And it will be difficult to assure that no particles are skipped during collision calculations, so the entire program might have to be restructured.