**Introduction**

***Context***

This report is a supporting document of the MATLAB model 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 in a vacuum chamber on a target material, the high energy density of the laser causes the atoms in the target surface 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 and reflect off the surface. The background gas increases the number of collisions between particles which reduces the total kinetic energy of the plasma, resulting in a higher absorption rate for particles arriving at the substrate. 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 of the plasma species. 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 cutting-edge methods, and is particularly suitable for development of complex oxide materials. In recent years upscaling of the technique is being actively researched. 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 the chemical composition of the expanding plasma, which could provide a helpful tool for target selection and PLD growth parameter optimization.

**Model description**

Firstly, 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. However, 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, where collisions under all possible angles occur. Nonetheless, the model is able to achieve results closely matching experiment. On average a similar number of particles exit and enter each angular bin each timestep, which could explain why this approximation is valid. 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, and could even be expanded to a 3D model.

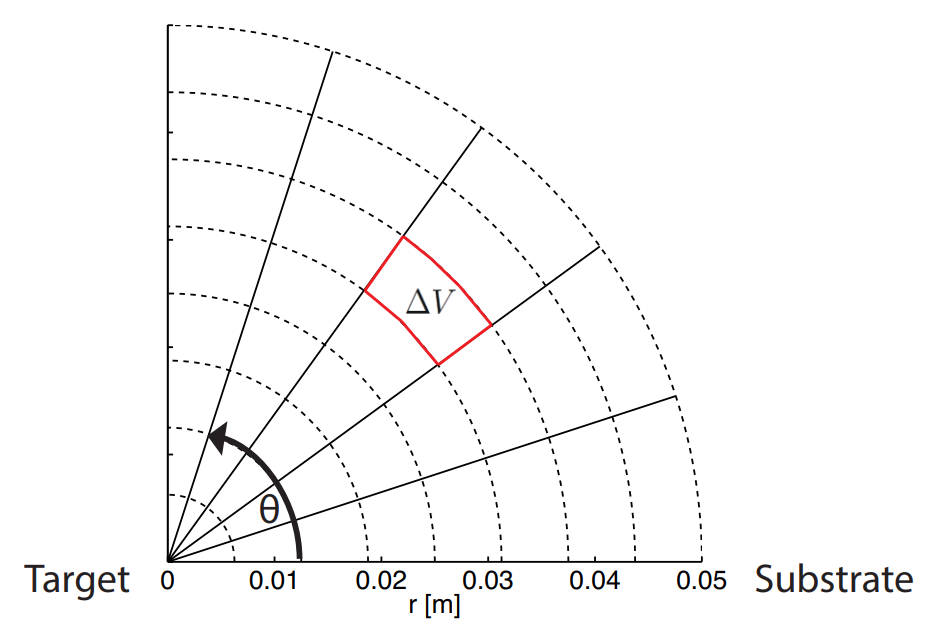


Figure 1 - Schematic of the polar grid which represent the spatial computational bins of the numerical model, where the horizontal axis shows the distance between target and substrate.

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 expected 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 negligible. 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 :

Not all ablated particles will have exactly the same velocity, instead they will initially follow some velocity distribution. 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 a Ti plasma originating from a TiO2 target. 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, according to Graham’s law an inversely proportional relation is expected between the distribution width, which represent the diffusion rate, and the particle mass, with a wider distribution for a lower mass and vice versa. So, the Gaussian distribution was replaced with a Maxwell-Boltzmann distribution 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:

Figure 2 shows an example of the initial velocity distribution of particles ablated from a Li4Ti5O12 target.

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 a single particle has experienced, which is initially zero. is the number of atoms of type per unit cell.

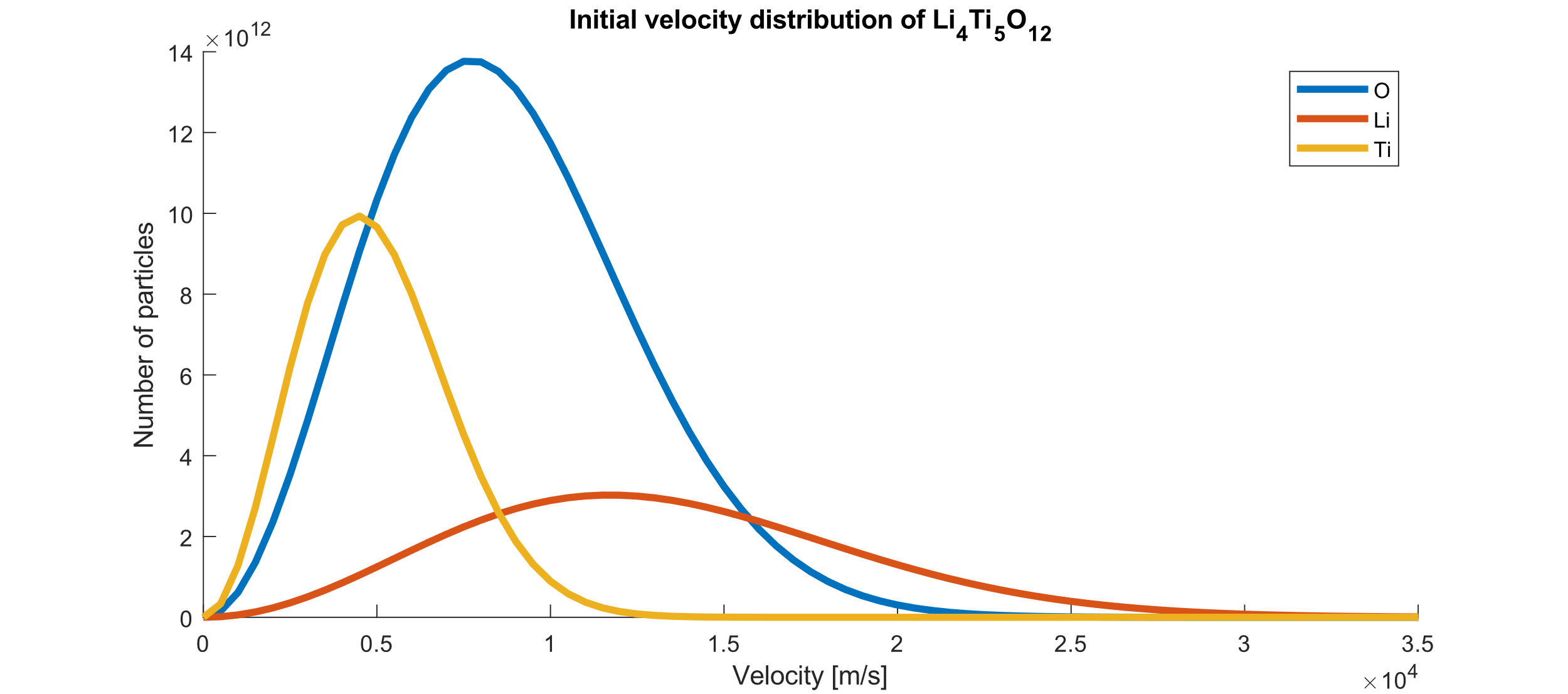


Figure 2 - Initial velocity distribution corresponding to the ablation of a Li4Ti5O12 target, following a Maxwell-Boltzmann distribution.

***Background gas particles***

For the results presented here the initial temperature of the background gas is assumed to be constant throughout space with . In a PLD chamber with a heated substrate 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. 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 set 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 approximations of the model is that all collisions are assumed to be head-on fully elastic collisions. Five distinct collision categories can be identified: (1) elastic collisions between the plasma and the background gas particles, (2) oxidation 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. Category 1 collisions are the only collisions included in this model. Category 2 oxidizing collisions are supported in the code, but have not been implemented yet. Category 3 collisions 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 4 and 5 collisions are not included due to time constraints and computational restrictions, but could potentially be implemented in the future.

Wijnand argued that category 4 and 5 collisions could be safely neglected if their initial average velocities were far apart, as this would result in different mass plasma particles separating in space. However, only with a normal distribution the overlap of velocities of different particles is minimal. If instead a Maxwell-Boltzmann, or even a log-normal distribution is used, then the overlap is much greater, and plasma particle collisions are expected to be significant. Especially, because the particle density in front of the target right after ablated is much higher than the background gas particle density. So, to obtain accurate physical results, these collisions have to be included in the model in future iterations.

***Number of collisions***

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

Here is the number of remaining non-collided particles from the previous radial bin, and is the collision rate between 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. (see figure 3) 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, which is defined as:

***Velocity and position after collision***

The velocity of two objects after a head-on elastic collision where momentum and kinetic energy are conserved can be calculated as:

The distance travelled by a collided particle within one time step is then determined by an average between the velocity before collision and after collision, weighted by how far the particle travelled in this time step before a collision occurred:

Where is the distance travelled before a collision occurred, and is the distance the particle would have travelled is no collision had occurred.

***Order of operations***

To ensure no particles get processed twice in one time step, the collided particles are first removed from their old positions. Then the position of the remaining non-collided particles is updated to their new position. And lastly the collided particles get added back with their new velocity and position described above.

The model can be summarized as follows:

1. Loop through all radial bins that are filled with plasma particles.
2. Loop through all velocity bins that are filled with plasma particles.
3. Store the number of plasma particles in this bin.
4. Calculate the travelled path based on the velocity of the particles in the case that no collisions occur.
5. Loop through the radial bins within the travelled path.
6. Loop through all velocity bins that are filled with background particles and that move slower than the plasma particles.
7. Calculate the number of collided particles.
8. Calculate the new velocities of plasma and background gas particles after collision.
9. Calculate the new positions of the plasma and background gas particles after collision.
10. Remove the collided particles from their velocity and radial bins prior to the collision.
11. Update the position of non-collided particles.
12. Add collided particles back to their new velocity and radial bins after collision.

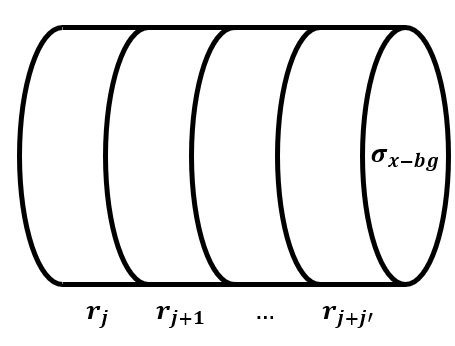


Figure 3 - Schematic view of the 'collision path volume'.

**Implementation**

The code and its corresponding classes and functions have been completely rewritten from the ground up. The data structure used to store all particles in the model has been 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 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.
* A separate data structure was use for the background gas, which is not very flexible and does not allow the full modelling of background gas particle.
* Particles could have any number of velocities. Arrays with variable length tend to be 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. Instead, the number of velocities should be compatible with the length of the radial bins and the duration of one time step, so travels one per , travels per , etc. Any additional resolution in velocity is a waste of resources.
* 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 using MATLAB’s convenient workspace viewer a hassle. Additionally, 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 radial positions in the struct 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.

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 4 shows a schematic overview of how the particle matrix is stored in memory. The main technical differences are as follows:

* The data structure is flexible, allowing it to be dynamically extended for any number of target species without exaggerated performance impact.
* The background gas particles can be fully modelled together with the plasma particles. In Wijnand’s model the background gas particles only had two velocity bins: one for static particles, and one for kinetic particles. In the new model the background gas can have every velocity. Also, the number of collisions of the background gas particles can be tracked.
* 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.
* 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.

Classes were created that hold atom and material unit cell properties. Additional atoms and materials can be easily added. The code will is completely dynamic, meaning that in theory every possible material can be added and simulated, as long as it fits the model assumptions and approximations.

**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 4 - Schematic view of how the data structure is stored in memory. N and C stand for non-collided and collided respectively.

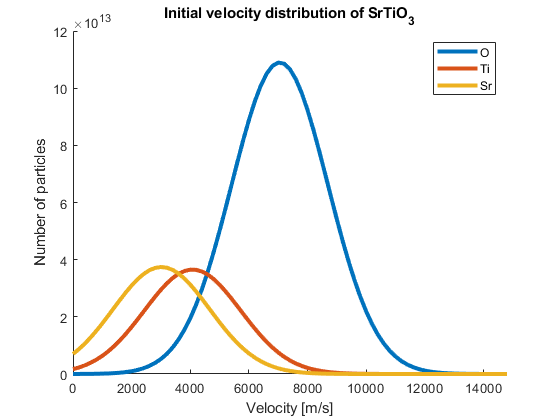
**Results and discussion**

NOT COMPLETE, SEE PRESENTATION FOR MORE RESULTS!

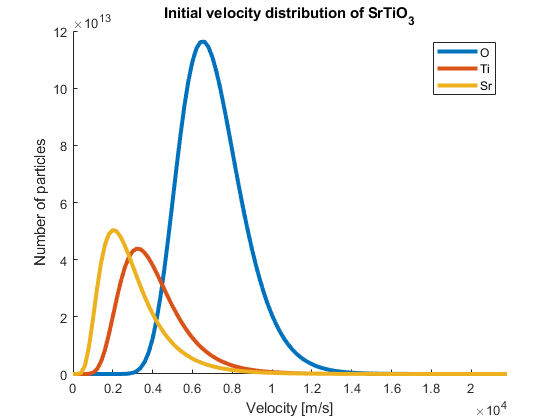
***Initial velocity distribution***

The initial velocity distribution was the first code that was redesigned. There was a mistake in Wijnand’s calculations that was corrected; the laser energy was divided by the ratio of energy absorbed by the target instead of multiplied, resulting in a higher average particle velocity. As described above, a system was developed in which materials can be dynamically changed and new materials can be easily added. Also, targets consisting of a mixture of multiple phases are supported if the ratio between the phases is known.

As mentioned before, the velocity distribution was also altered. The normal distribution used in Wijnand’s paper extends into negative values for heavy atoms with low initial mean velocity, as shown in the figure below. If all particles are assumed to be fully energized by the laser, particles with zero or negative velocity are impossible immediately after ablation.

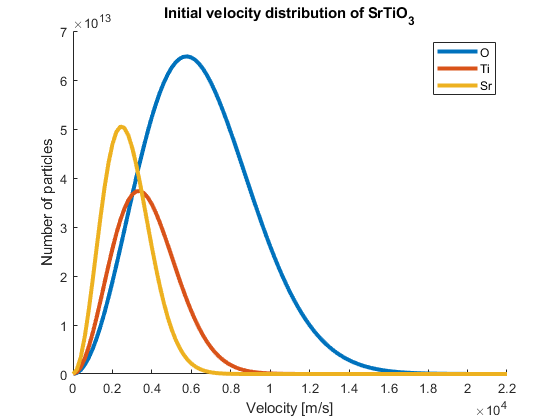


The first idea was to use a log-normal velocity distribution, which spans , and is always zero at zero velocity. For higher mean velocities, a log-normal distribution approximates a normal distribution. With this distribution the same velocity distribution width that Wijnand found for TiO2 can be used.



However, it is actually not expected that particle of different mass have the same velocity distribution width. According to Graham’s law, an inversely proportional relation is expected between the distribution width and the particle mass. Also, Wijnand based his distribution width on spectral images made of the expansion of a TiO2 plasma. As oxygen is too light to measure with spectroscopy, these images only show the expansion of titanium and titanium oxides. The assumption that the same distribution width can be used for the expansion of oxygen is not based on experiment.

Alternatively, a more physically motivated distribution is the Maxwell-Boltzmann distribution. This distribution only spans positive value, like the log-normal distribution, and it gives a physical approximation of the distribution width inversely proportional to the atomic mass.



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. Note that 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.

Using this distribution the oxygen and metals will not separate in spaced as Wijnand argues would happen in the case of a normal distribution. Indeed, there is significant overlap in their velocity distributions. This makes the assumption that the interactions 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.

**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 collisions are head-on and fully elastic.
* 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.

***Improvements compared to Wijnand’s model***

* Fully model background gas particle kinematics.
* Support 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.
* 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.
* The code is self-documenting and contains no magic variables. All user input can be found at the top of the script with clear comments.

***Features missing compared to Wijnand’s model***

* Oxidation of plasma species
* 2D plume expansion plots

***Suggested 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.
* Finding an approximation for the heat dissipation into the target.
* 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 highly kinetic particles to reflect of the substrate surface and change direction.
* The order of operations of the model has to be altered. Now particles get updated in order of their location in space. In reality everything happens at once, but a computer can only compute one thing at a time, so a better order of operations has to be thought of. If we assume that faster moving particles have a higher collision probability, it makes sense to update the particles in order of their speed instead of their location.
* Keeping track of number of collisions per particle (or in this implementation collided and non-collided) is mainly for plotting purposes, as it better visualizes which particles have interacted and which have not. Alternatively, the plots could be separated in highly kinetic particles, and less kinetic particles. For example, the initial RMS velocity could be used as threshold for if a particle is considered highly kinetic or not. Additionally, it could be useful to plot particles with negative and zero velocity separately to visualize their propagation direction.
* Try to divide space into equal volume bins instead of equal distance bins. If there is radial segments, with being the final radial segment, then segment is located at:

This ensures equal volume for each spatial bin. The downside of this approach is that no longer each velocity exactly maps toa distance in radial bins, so travels one radial bin per timestep, and travels two radial bins per timestep, etc. This makes the choice of resolution in time, space and velocity not as clear cut.

***Proposal for an implementation of including collisions between all species***

Let us consider the plasma propagating in a vacuum. When using a Maxwell-Boltzmann initial velocity distribution function there is significant overlap of velocities between different plasma particles. The particles whose velocity distribution overlap do not separate in space, and within this area they can undergo collisions with every particle with a lower speed or opposite propagation direction. Now we introduce a background gas. At deposition pressures (roughly 0.01-0.5 mbar) the background gas particle density directly in front of the target is orders of magnitude lower than the plasma particle density right after ablation. This means that in the initial moments after ablation, collisions between plasma species have a much higher probability than collisions with the background gas.

These arguments summarize why the approximation of only modelling collisions of plasma species with the background gas is not valid in the early stages of ablation, and might only give reasonably results in the later stages of plasma expansion where the local plasma density is of the same order as the local background gas density.

If collisions between all particles are included, the number of collisions per individual plasma particle is expected to become high, so tracking the number of collisions per particle might not be as useful anymore for the plasma particles, as every plasma particle is expected to collide many times during the first moments of expansion where the very high particle densities occur. However, it is still useful for separating the background gas particles in number of collided and non-collided particles, so its easily visible which part of the background gas is energized. Alternatively, this could be achieved by plotting the zero and non-zero velocity particles separately. If the number of collisions per individual particle is not important anymore, the model can be reduced with one dimension, so an 3D matrix remains.

To implement collisions between all particles, the model has to be restructured, as the current collision calculation method is too slow and requires millions of loops. The following method was envisioned:

* Loop through all particle speeds
  + Loop through the two particle propagation directions (towards the substrate or towards the target)
    - Determine the path the particle with this speed and direction will travel if no collisions occur.
    - Loop through all radial bins
      * Limit the travelled path by the radial boundaries. (target and substrate surface)
      * Loop through all plasma species
        + Get the number of particles of this plasma species, in this radial bin, at this velocity bin.
        + Skip the loop if the number of particles if less than the threshold.
        + Get the number of particle of all other species (including the background gas) that are not this species, or the one of the species that has already been looped through. Get it for all velocities slower of opposite to this velocity. Get it for all radial bins in the travelled path.
        + Skip the loop if all the bins contain a number of particles less than the threshold. (this line should be checked how likely it is to occur and if it actually saves time)
        + Loop through the radial bins in the projected path

Calculate the number of collisions per bin

Update the number of particles of this species that is left to collide with by subtracting the number of collisions.

Break loop if the number of particles of this species is less than the minimal number of particle threshold.

* + - * + Skip loop if all number of collisions are below the number of particle threshold.
        + Add collided particles with their velocity and position before collision to a subtraction matrix.
        + Calculate the velocity and position after collision.
        + Loop through all other species

Add respective number of collision to an addition matrix.

* + After looping through all particles with this speed, check if the number of collisions in any bin is higher than the number of particles that is present in that bin. Limit these bins by the number of particles in that bin to prevent negative number of particles.
  + Add all collisions to a total number of collisions sum.
  + Subtract the subtraction matrix from the particle matrix, so collided particles are not available for collisions with particles moving at a lower speed, to prevent particles being updated or collided twice in one time step.
* Update the positions of the remaining particles that have not collided this timestep.
* Normalize the addition matrix by the total number of collisions sum, to ensure an equal number of particles get removed and added each time step.
* Add the addition matrix holding all the collided particles and their new positions and velocities back to the particle matrix.

An attempt at implementing this method is included in alternativeSolver, but it was not working correctly yet. Things to consider:

* It might be possible to directly add back collided particles after looping when updating per particle velocity. Although, this will result in multiple collisions occurring for one particle in one time step. The difference between adding back particles after each speed loop versus after looping through every velocity should be investigated.
* It might not be necessary to update the particles position during collision calculation, and just updating the velocities, and then just updating the positions based on their new velocities could suffice. This will result in the particle moving slightly less far each timestep after they underwent a collision.
* Implementing this with variable radial bin length, but constant spatial bin volume will result in some difficulties. For example, the highly efficient updating method in updateMatrix using bit shifting cannot be used anymore. It could potentially be rewritten with variable bit shift distance based on the position and velocity. But probably the older less efficient looping methods included in update2DMatrix and update2DMatrix have to be used.