# PLD Plasma Propagation Model

Based on a model by Tom Wijnands

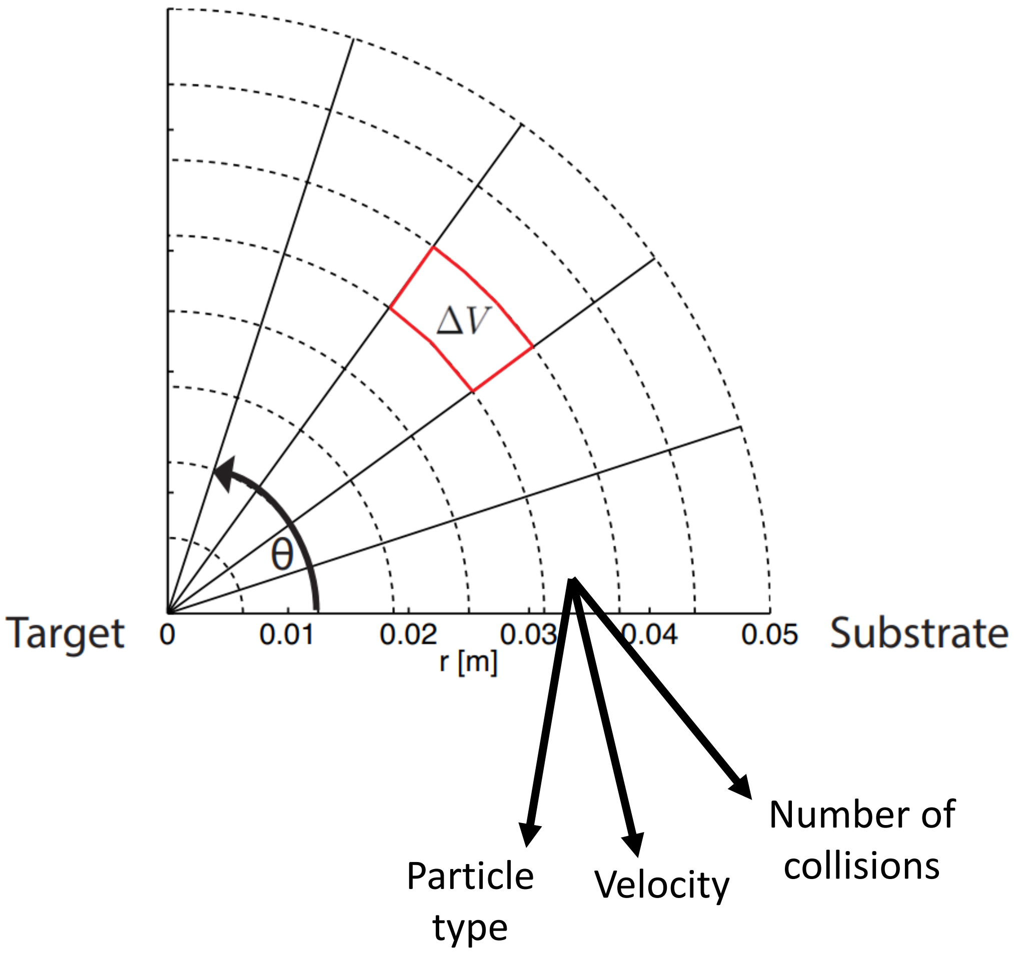
Adapted by Sam Borkent

# Introduction

- Model description
- Approximations and assumptions
- What did I do and why?
- Results
- Conclusions
- Suggestions
- Let’s discuss

# Model description

- **Divide space into computational bins**
- Determine the number of ablated particles per laser pulse
- Calculate the initial velocity of particles immediately after ablation
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- Perform a 1D collision calculation problem for every angle
- Combine 1D results to achieve a 2D propagation model



# Model description

- Divide space into computational bins
- **Determine the number of ablated particles per laser pulse**
- Calculate the initial velocity of particles immediately after ablation
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- Perform a 1D collision calculation problem for every angle
- Combine 1D results to achieve a 2D propagation model

$$N_{p,tot} = N_{p,uc} \frac{V_{spot}}{V_{uc}} \frac{\rho_{target}}{\rho_{sc}}$$

$N_{p,uc}$: number of particles per unit cell

$V_{spot}$: volume of the ablation spot

$V_{uc}$: volume of a unit cell

$\rho_{target}$: density of the target

$\rho_{sc}$: density of a perfect material

# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- **Calculate the initial velocity of particles immediately after ablation**
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- Perform a 1D collision calculation problem for every angle
- Combine 1D results to achieve a 2D propagation model

$$E_{laser} = E_{reflected} + E_{absorbed}$$

$$aE_{laser} = Q + N_{uc}\left(E_b + \sum_{x=1}^{N_{p,uc}} \left(E_{kin,x} + \Delta E_{exc,x}\right)\right)$$

$$\bar{E}_{kin,x} = \frac{1}{N_{p,uc}}\left(\frac{aE_{laser} - Q}{N_{uc}} - E_b\right) - \Delta E_{exc,x}$$

$\alpha$: ratio of laser energy absorbed by target

$Q$: heat dissipation into the target

$N_{uc}$: number of particles per unit cell

$E_b$: atomization energy (not formation energy)

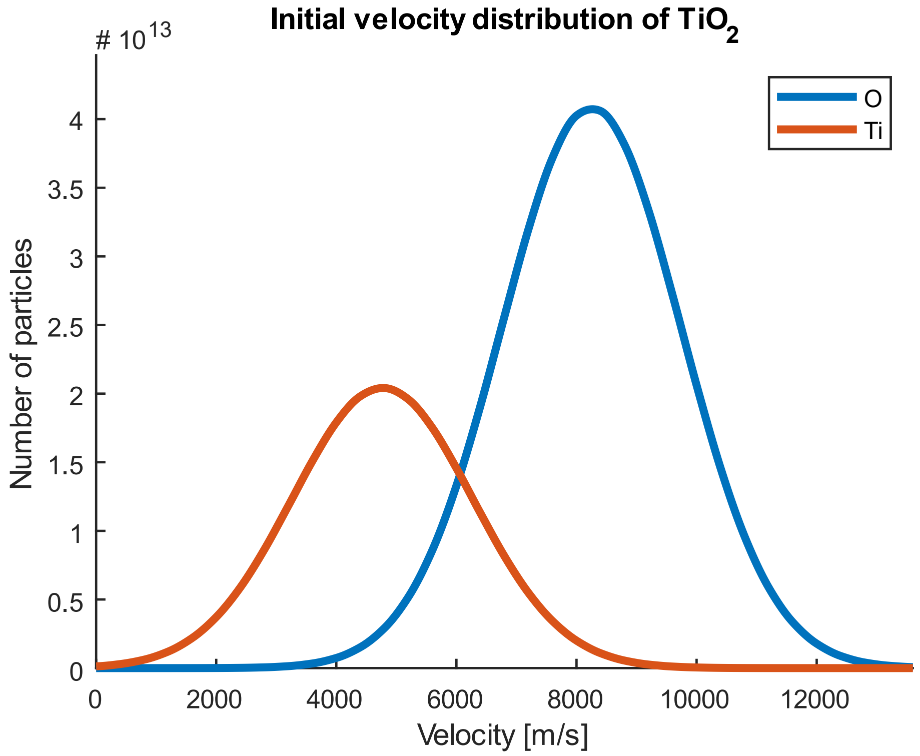
$\bar{E}_{kin,x}$: average kinetic energy of atom of type $x$

$\Delta E_{exc,x}$: average excitation energy of atom of type $x$

# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- **Calculate the initial velocity of particles immediately after ablation**
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- Perform a 1D collision calculation problem for every angle
- Combine 1D results to achieve a 2D propagation model

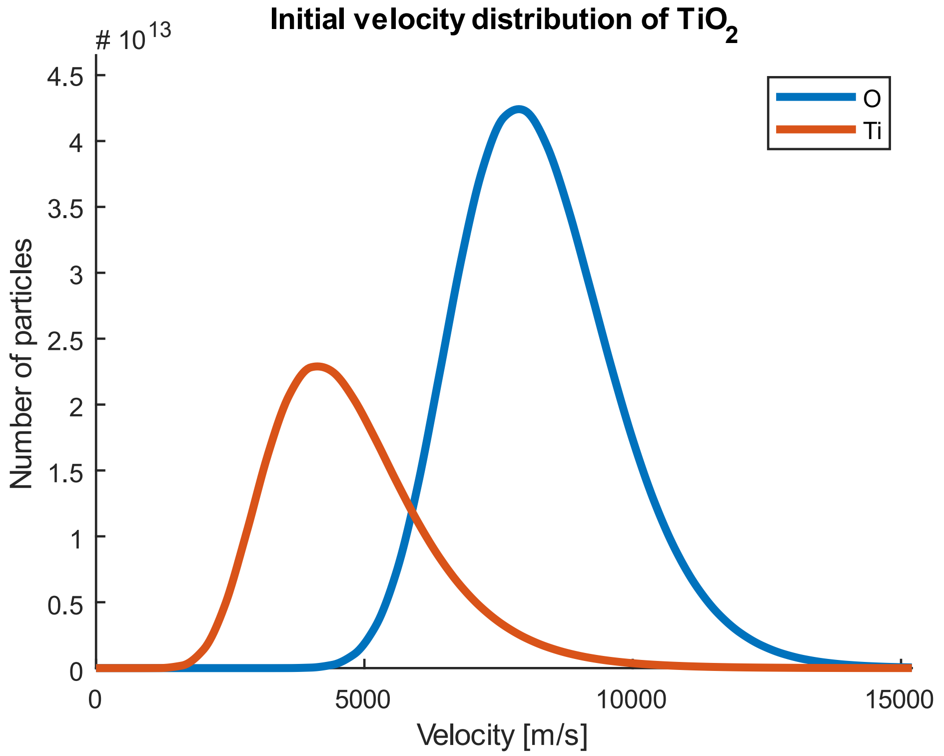
## Normal distribution



# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- **Calculate the initial velocity of particles immediately after ablation**
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- Perform a 1D collision calculation problem for every angle
- Combine 1D results to achieve a 2D propagation model

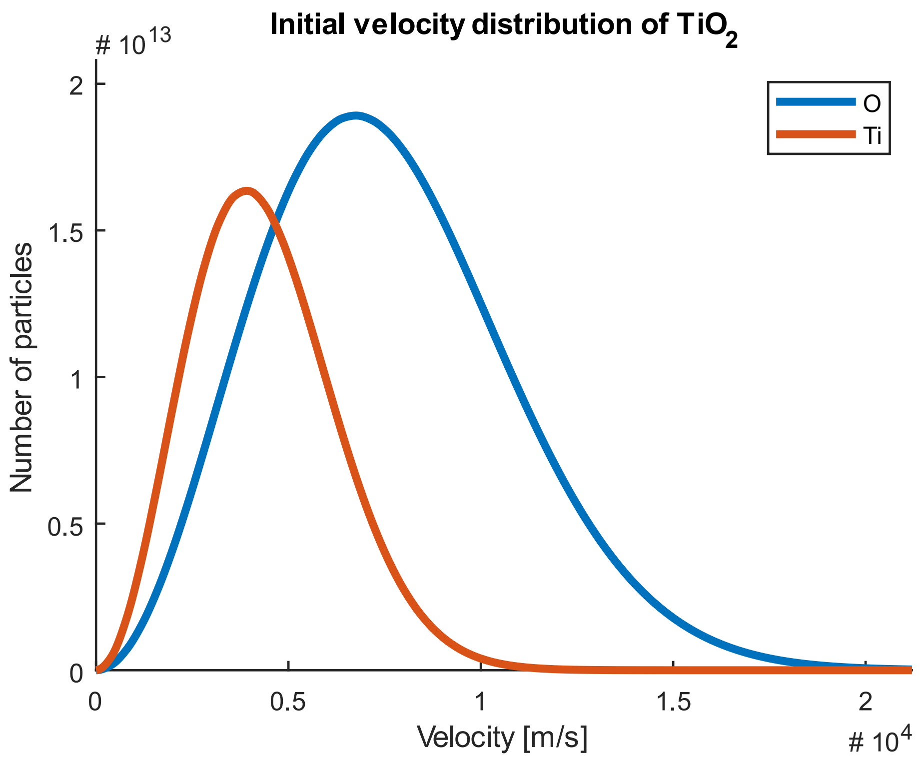
## Log-normal distribution



# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- **Calculate the initial velocity of particles immediately after ablation**
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- Perform a 1D collision calculation problem for every angle
- Combine 1D results to achieve a 2D propagation model

## Maxwell-Boltzmann distribution



# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- Calculate the initial velocity of particles immediately after ablation
- **Determine the number of background gas particles based on the deposition pressure**
- Collect the plasma and background particles into their designated computational bin
- Perform a 1D collision calculation problem for every angle
- Combine 1D results to achieve a 2D propagation model

**Ideal gas law**

$$\rho_{bg} = \frac{P_{bg}}{k_B T}$$

**Equipartition theorem**

$$\bar{E}_{kin,bg}(t_0) = \frac{1}{2} m_{bg} \bar{v}_{bg}^2(t_0) = \frac{3}{2} k_B T_{bg}$$

**Computational bin volume**

$$\Delta V(r_j, \theta_k) = \frac{4}{3}\pi\left(r_{j+1}^3 - r_j^3\right)(\cos(\theta_k) - \cos(\theta_{k+1}))$$

**Number of background gas particles**

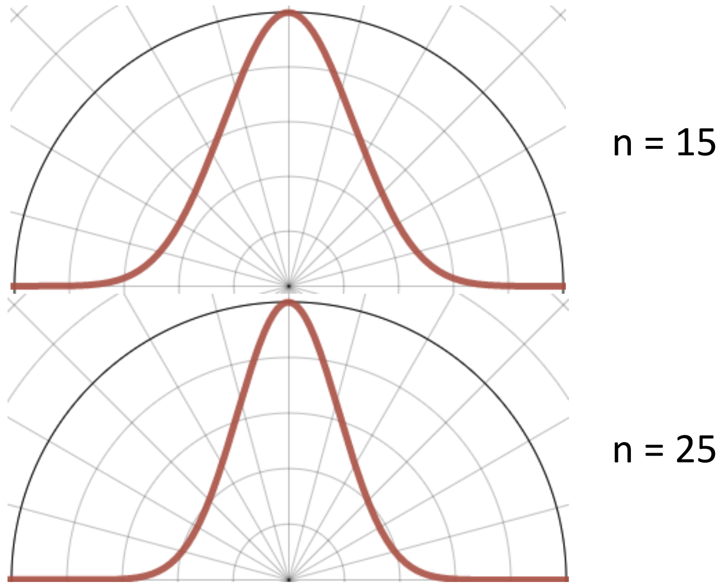
$$N_{bg}(r_j, \theta_k, t_0) = \rho_{bg} \Delta V(r_j, \theta_k)$$

# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- Calculate the initial velocity of particles immediately after ablation
- Determine the number of background gas particles based on the deposition pressure
- **Collect the plasma and background particles into their designated computational bins**
- Perform a 1D collision calculation problem for every angle
- Combine 1D results to achieve a 2D propagation model

**Angular distribution of plasma particles**

$$f(\theta) = \frac{\cos^n \theta}{A_\theta}$$




**Number of plasma particles**

$$N_x(n = 0, v_i, r_0, \theta_k, t_0) = N_{p,tot} \frac{N_{p,uc,x}}{N_{p,uc}} f_x(v_i) f(\theta_k)$$

# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- Calculate the initial velocity of particles immediately after ablation
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- **Perform a 1D collision calculation problem for every angle**
- Combine 1D results to achieve a 2D propagation model

**Collision categories**

- **Elastic collisions between particle of the same type**
- **Elastic collisions between the plasma and background**
- **Reactive collisions between the plasma and background**
- **Elastic collisions between metals in the plasma**
- **Collisions between metals and oxygen in the plasma**

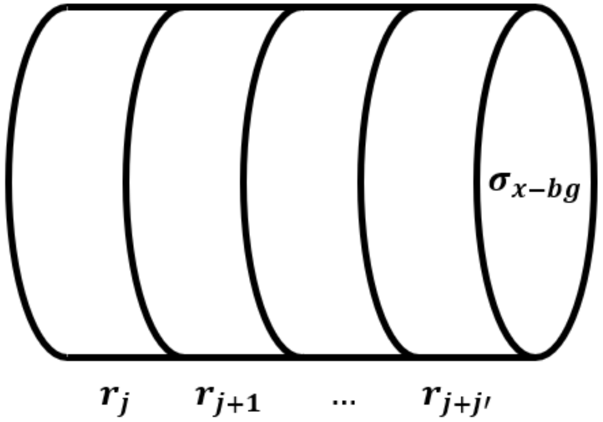
# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- Calculate the initial velocity of particles immediately after ablation
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- **Perform a 1D collision calculation problem for every angle**
- Combine 1D results to achieve a 2D propagation model

**Number of collisions in per bin**

$$N_{col} = N_p \rho_{bg} \Delta r \sigma_{x-bg} P_v(v_i, v_{i'})$$

$N_p$: number of plasma particles
$\rho_{bg}$: background gas particle density
$\Delta r$: length of one radial bin
$\sigma_{x-bg}$: collision cross-section
$P_v(v_i, v_{i'})$: relative velocity term




*"Collision volume"*

# Model description

- Divide space into computational bins

- Determine the number of ablated particles per laser pulse

- Calculate the initial velocity of particles immediately after ablation

- Determine the number of background gas particles based on the deposition pressure

- Collect the plasma and background particles into their designated computational bin

- **Perform a 1D collision calculation problem for every angle**

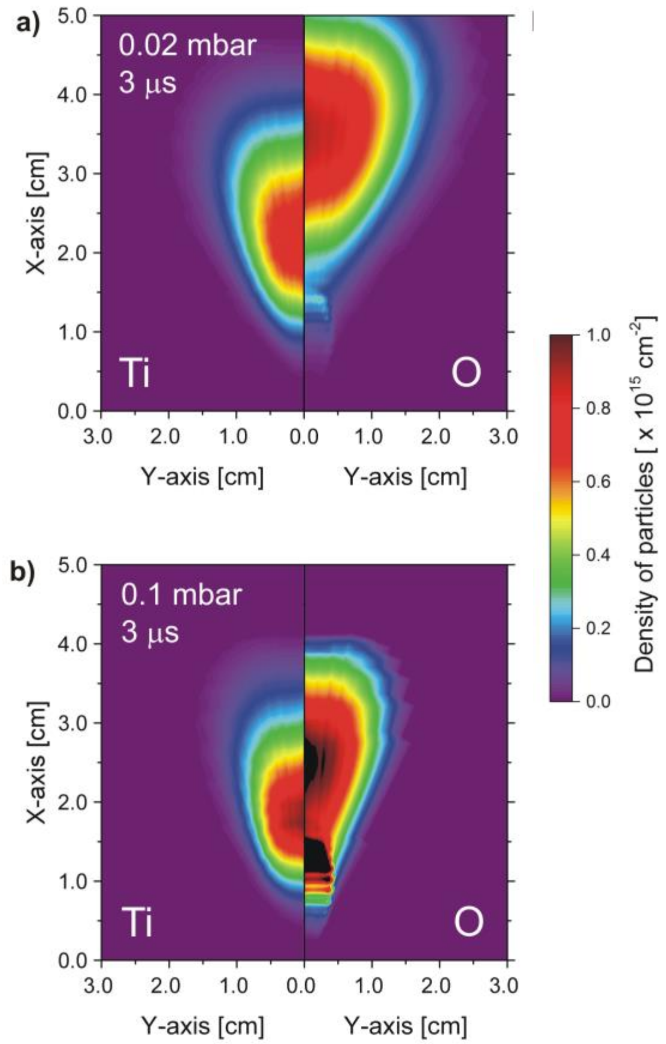
- Combine 1D results to achieve a 2D propagation model

**Collision calculation steps**

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

# Model description

- Divide space into computational bins
- Determine the number of ablated particles per laser pulse
- Calculate the initial velocity of particles immediately after ablation
- Determine the number of background gas particles based on the deposition pressure
- Collect the plasma and background particles into their designated computational bin
- Perform a 1D collision calculation problem for every angle
- **Combine 1D results to achieve a 2D propagation model**



# Approximations and assumptions

- There is no net exchange of particles between angular bins
- All particles move along a straight line following their initial angle
- Background gas particles initially have a constant temperature (no heat gradient)
- Background gas particles initially have no preferential propagation direction, so a net zero velocity
- Square laser ablation spot, resulting in an axially symmetric plasma expansion
- Heat dissipation into the target and excitation energy of atoms during ablation can be neglected
- All collisions are head-on and fully elastic
- Collisions only occur between plasma particles and background gas particles

## What did I do?

- Redesigned the code from the ground up
- Reimplemented the initial velocity distribution
- Implemented support for any target composition
- Fully model background gas particle kinematics
- Ensured conservation of number of particles, kinetic energy, and momentum

## Why did I do it?

- Make the model dynamic and suitable for any material
- Increase usability and readability of the code
- Improve performance

- Normal distribution is not the proper distribution to use for values that span $[0, \infty]$
- Velocity distribution width determined for the propagation of Ti, not applicable for other atoms

- Switch between materials by changing one line of code
- Easily add new materials

- Gain insight into how much oxygen gas is propelled onto the substrate surface

- No particles get destroyed or created

# GitHub (private)

| | | |
|---|---|---|
| classes | Updated initialVelocityDistribution | last month |
| documentation | Made some last adjustments, see README. | 2 minutes ago |
| functions | Made some last adjustments, see README. | 2 minutes ago |
| misc | Worked on alternative solver, fixed some bugs. | 4 days ago |
| results | Generated some results and investigated performance issues. | 22 days ago |
| README.md | Made some last adjustments, see README. | 2 minutes ago |
| alternativeSolver.m | Made some last adjustments, see README. | 2 minutes ago |
| plasmaSolver.m | Made some last adjustments, see README. | 2 minutes ago |

README.md

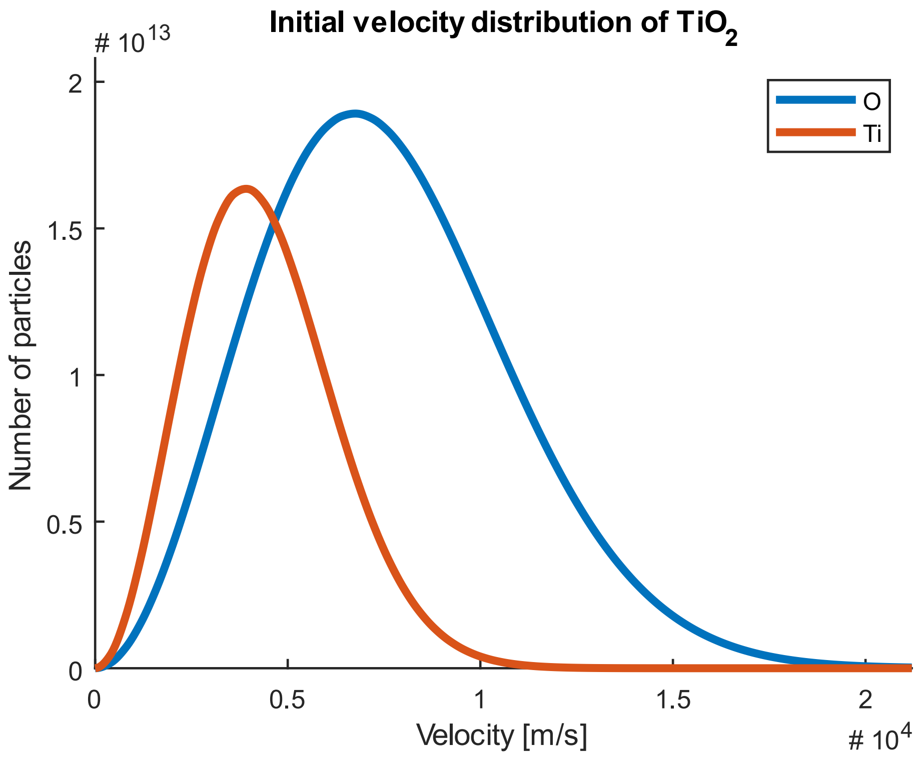
## Plasma Solver

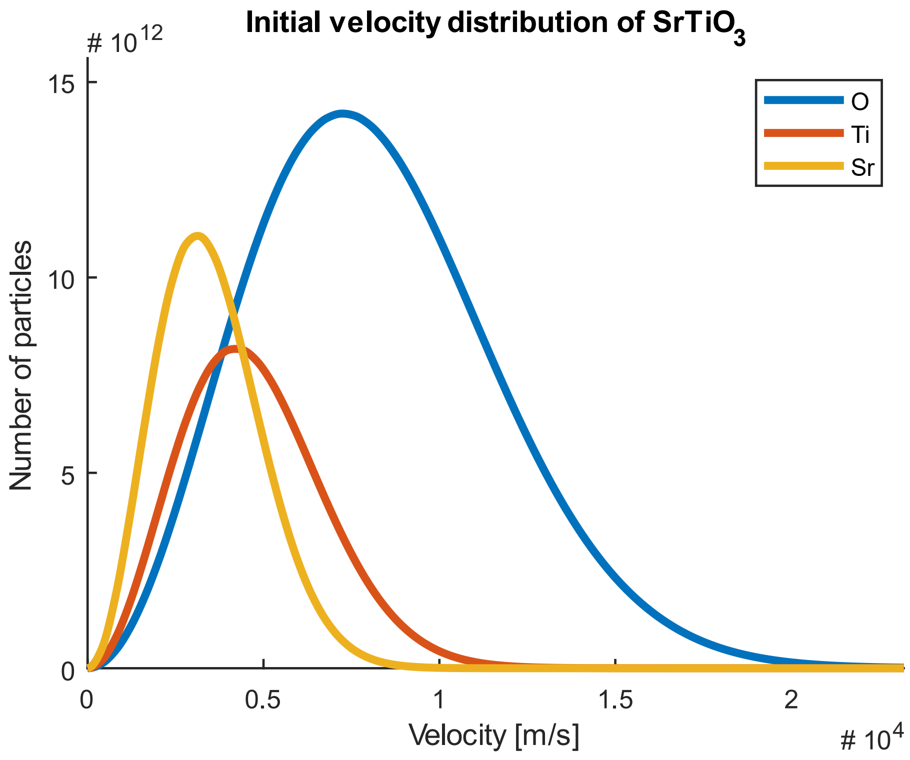
Author: Tom Wijnands, Sam Borkent

Based on 'Numerical modelling of the plasma plume propagation and oxidation during pulsed laser deposition of complex oxide thin films', 2020, by T. Wijnands, E.P. Houwman, G. Koster, G. Rijnders, and M. Huijben.

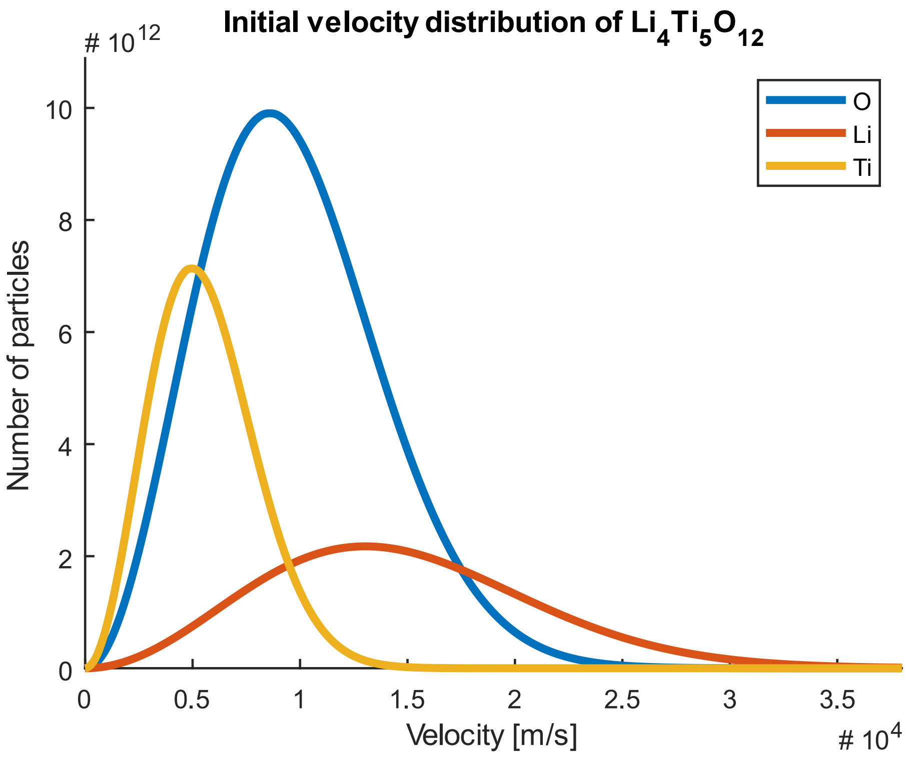
The original script by Tom Wijnands modeled the propagation of an PLD plasma plume in 2D as result of ablation of a single crystal TiO2 target, initial ablation is not included. Rewritten and extended by Sam Borkent to improve usability and performance and to support targets of any composition consisting of species of any mass.

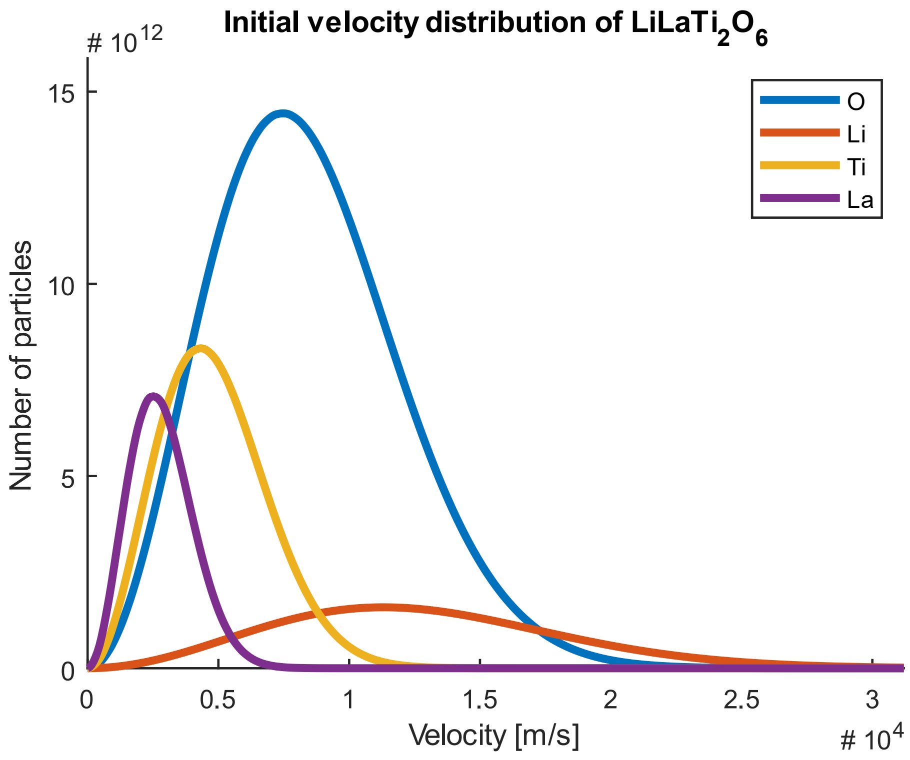
# **Results** Initial velocity distribution



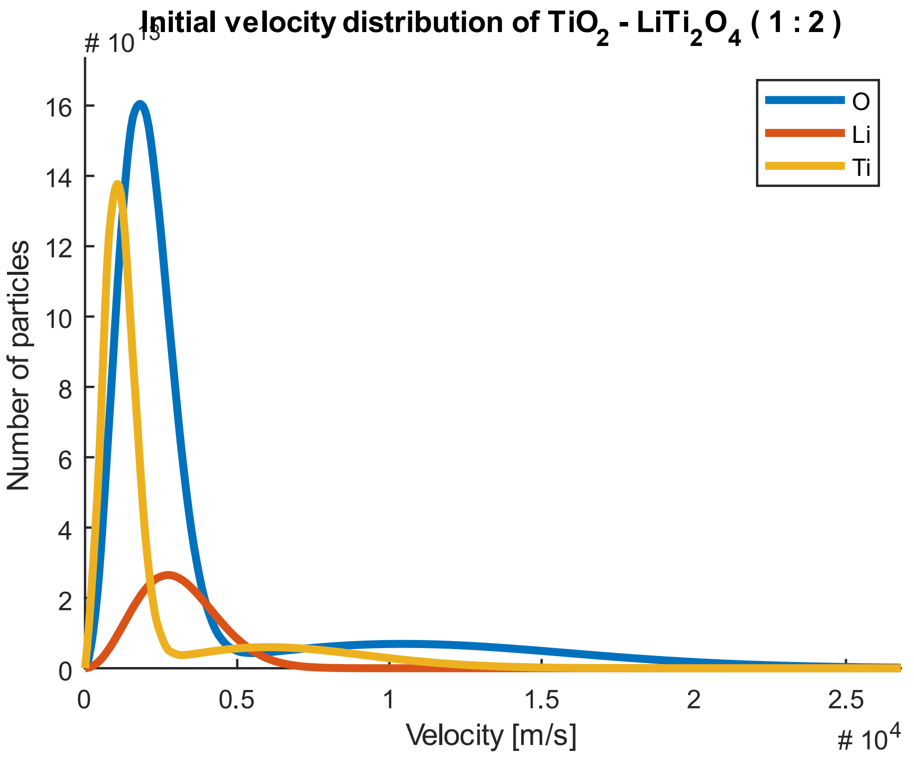



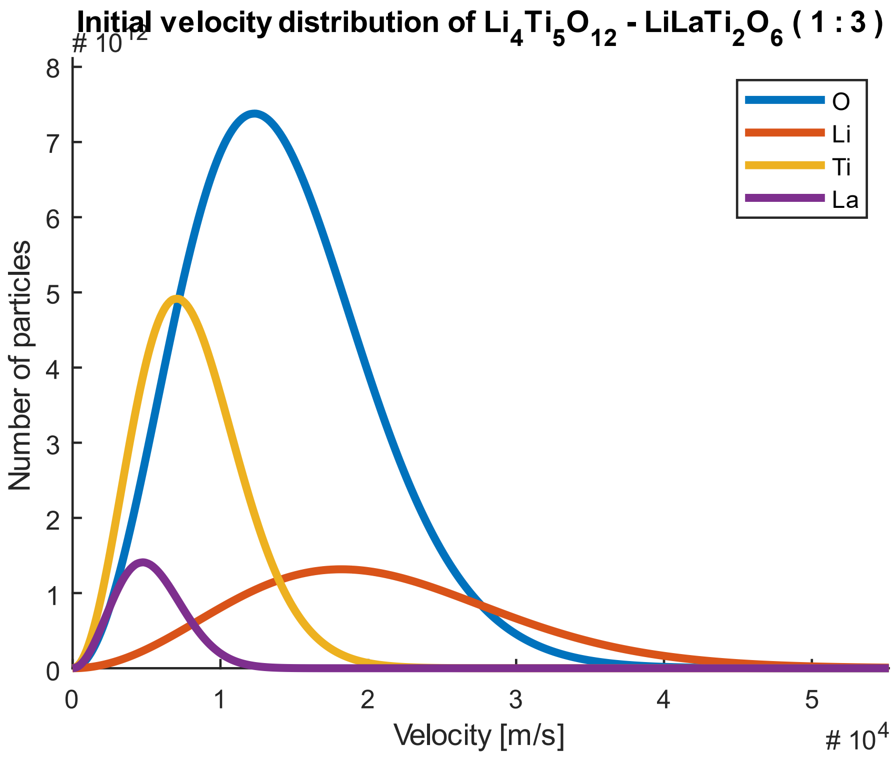
## **Results** Initial velocity distribution



**Results** Initial velocity distribution



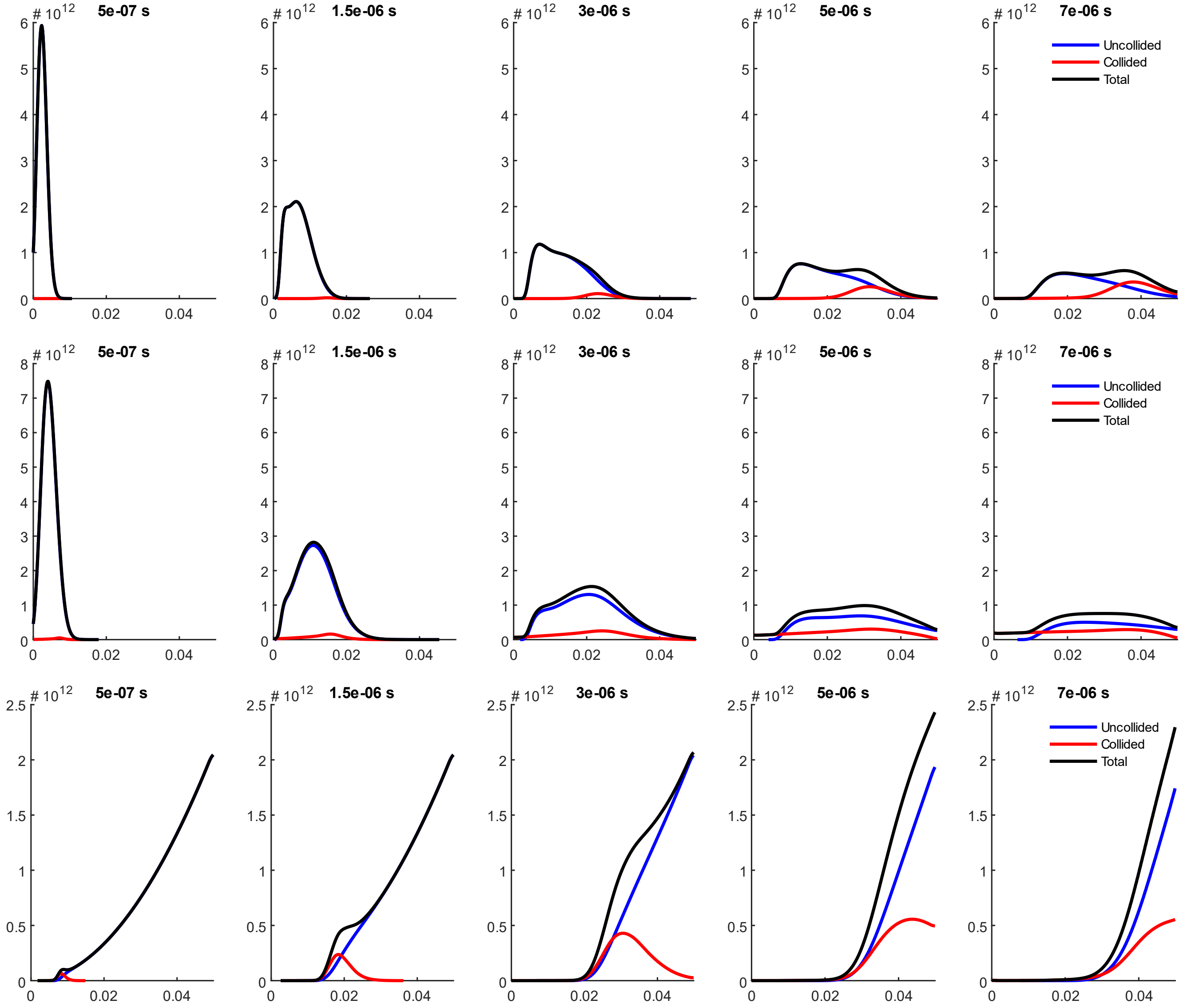



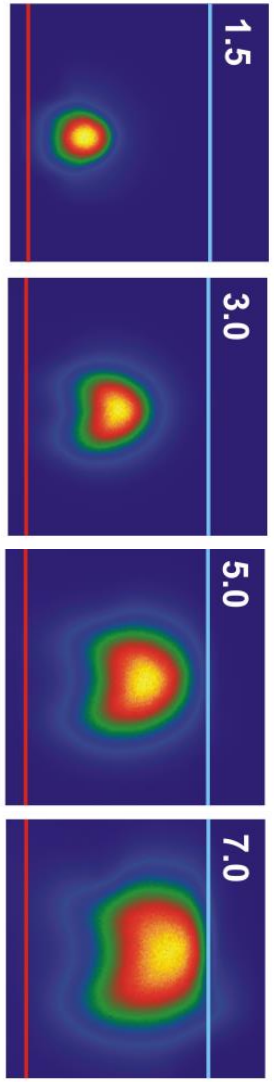
# Results Plasma propagation – TiO2 at 0.02 mbar

Ti

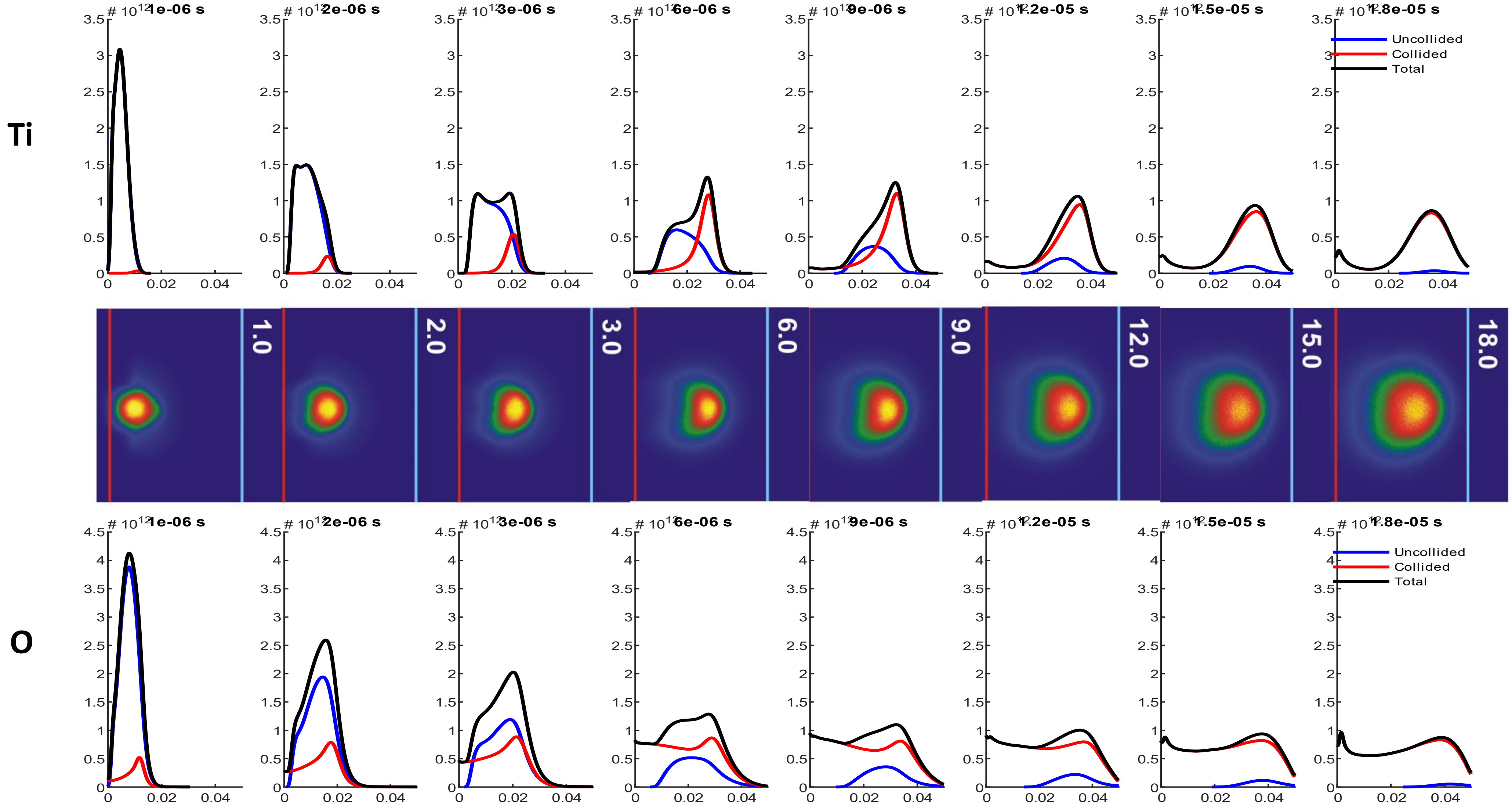
O

Bg O2

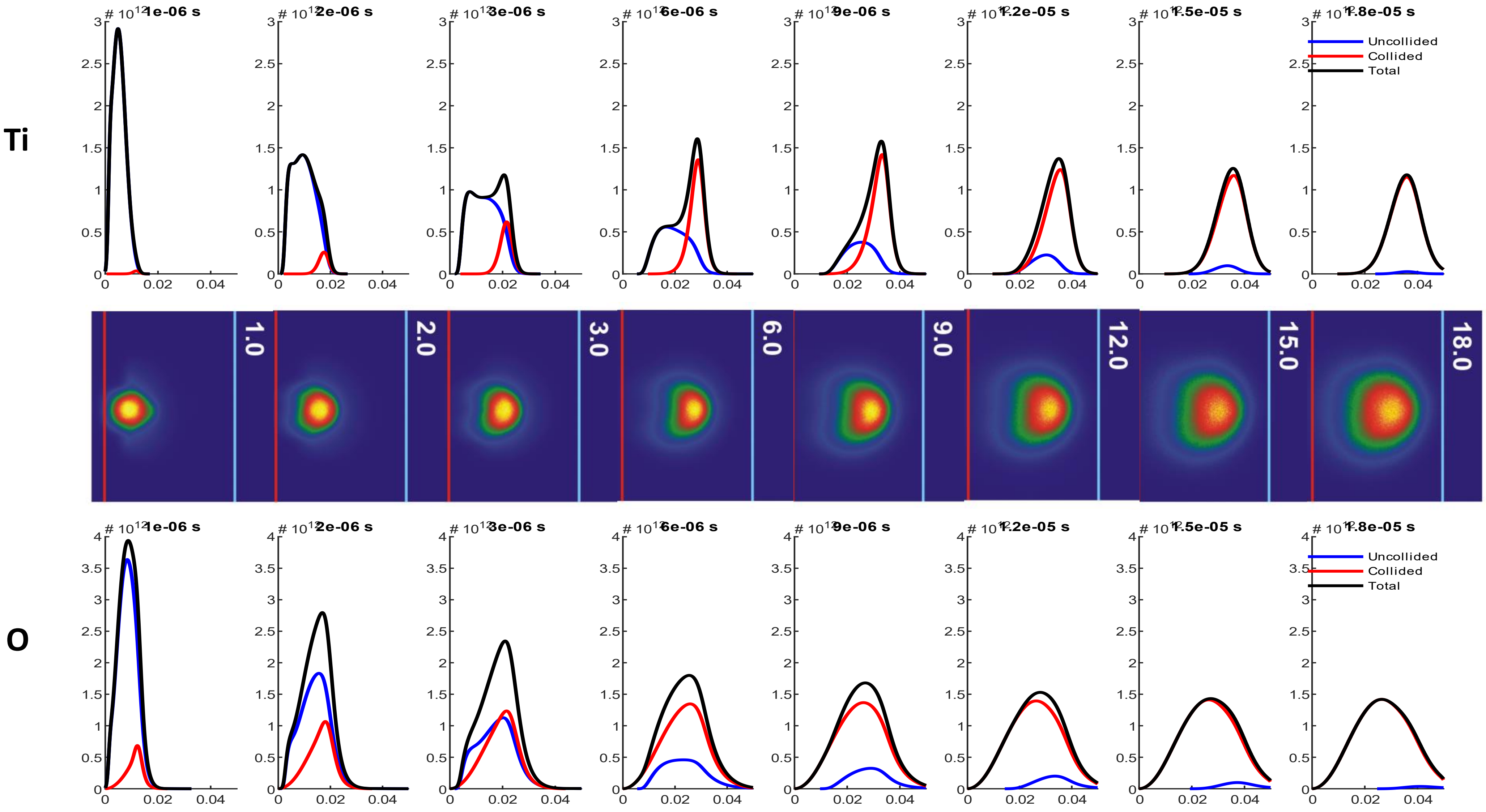


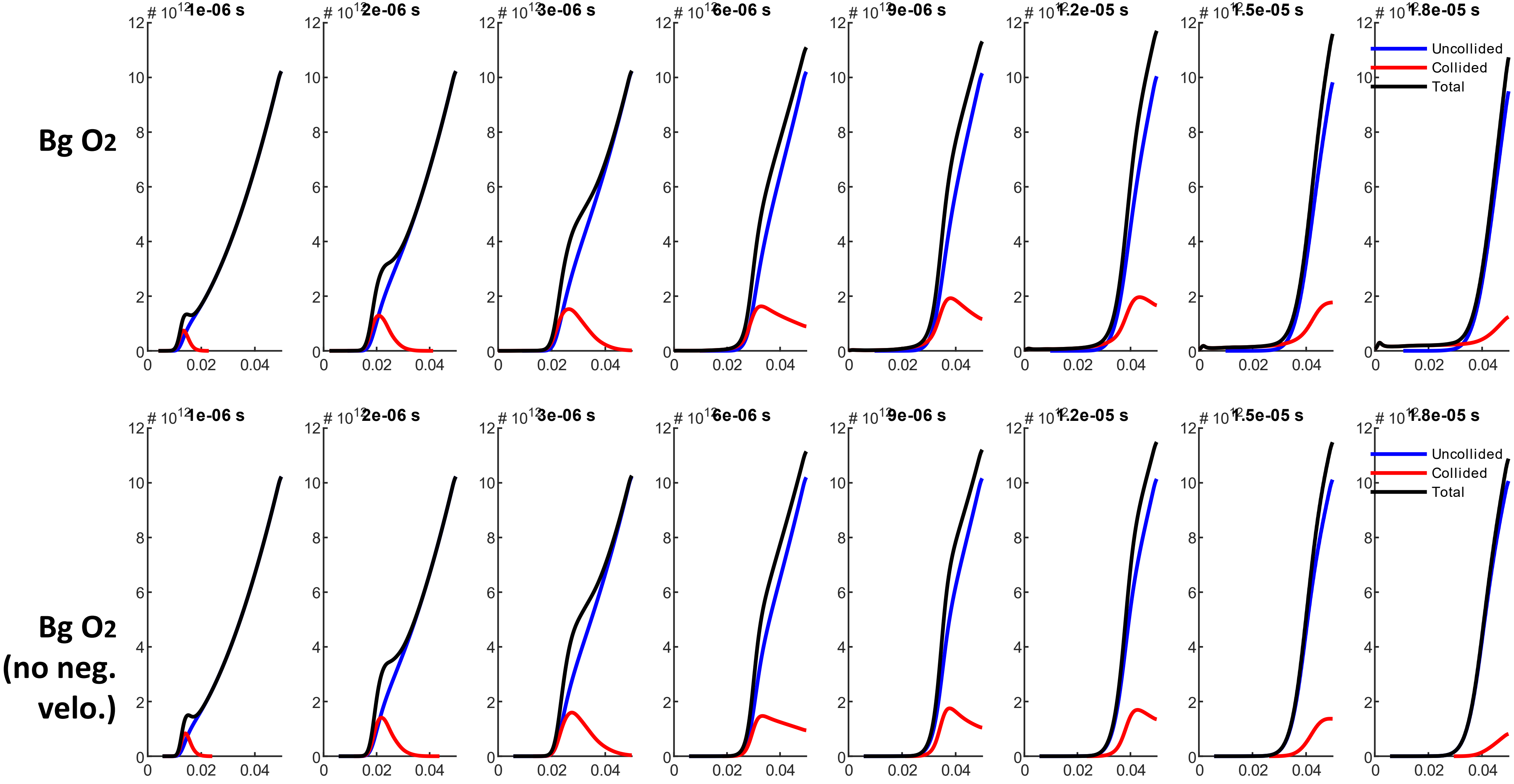
# **Results** Plasma propagation – TiO2 at 0.1 mbar



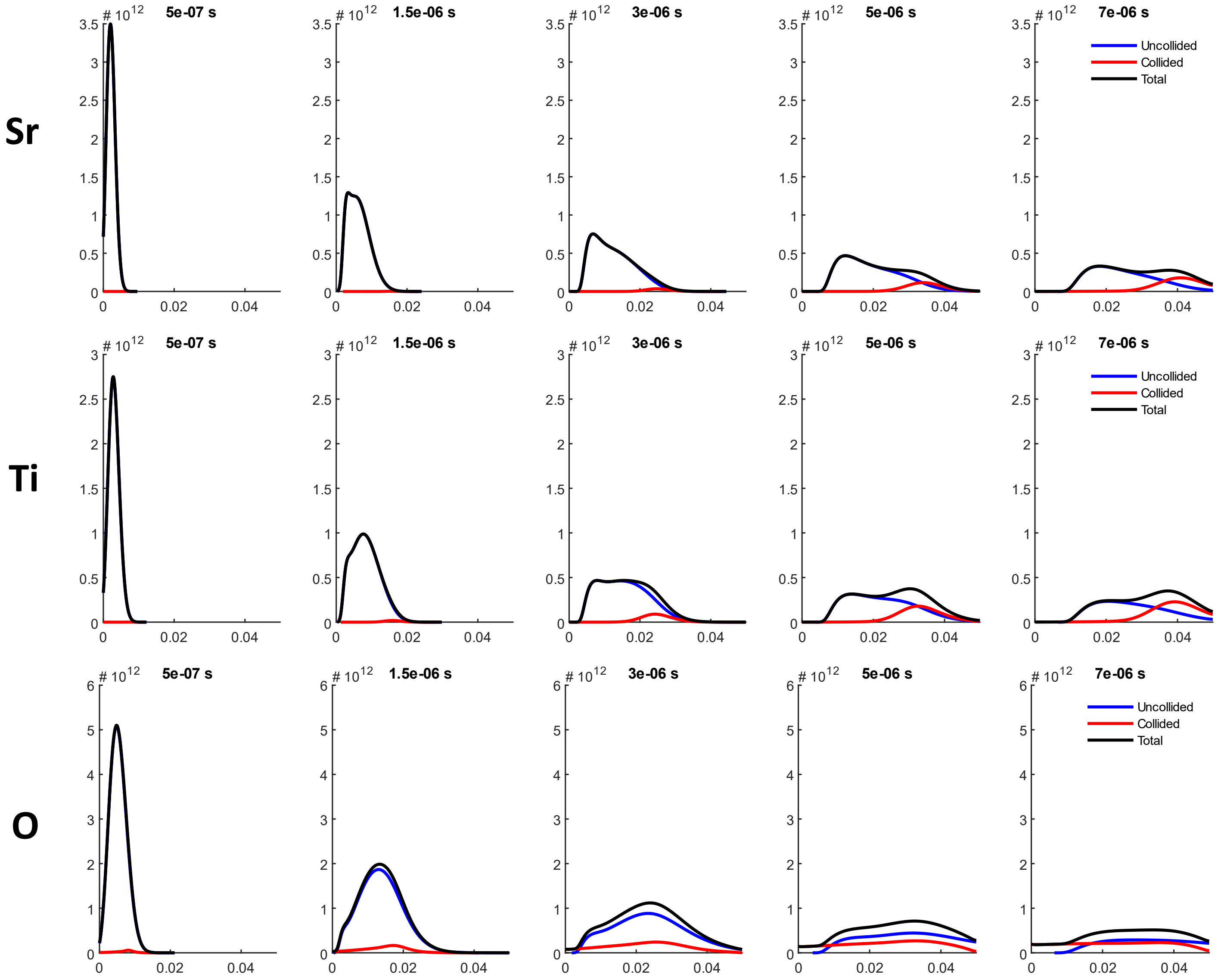
**Results** Plasma propagation – TiO2 at 0.1 mbar (no neg. velo.)

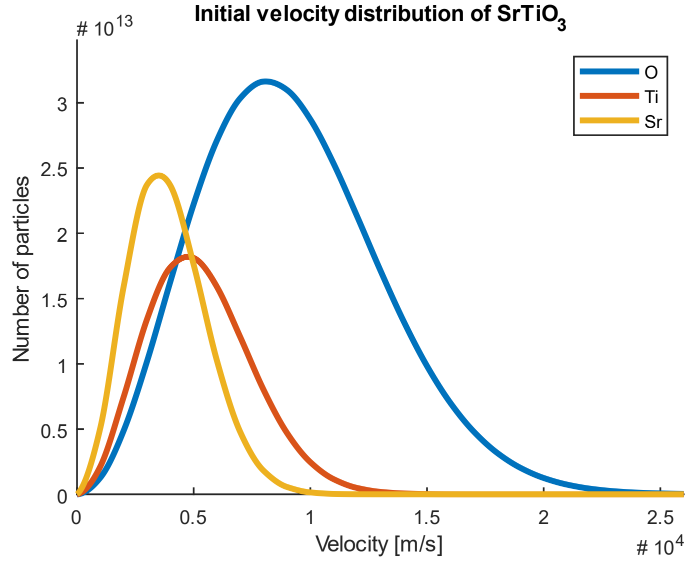


# **Results** Plasma propagation – $TiO_2$ at 0.1 mbar



# Results Plasma propagation – SrTiO3 at 0.02 mbar



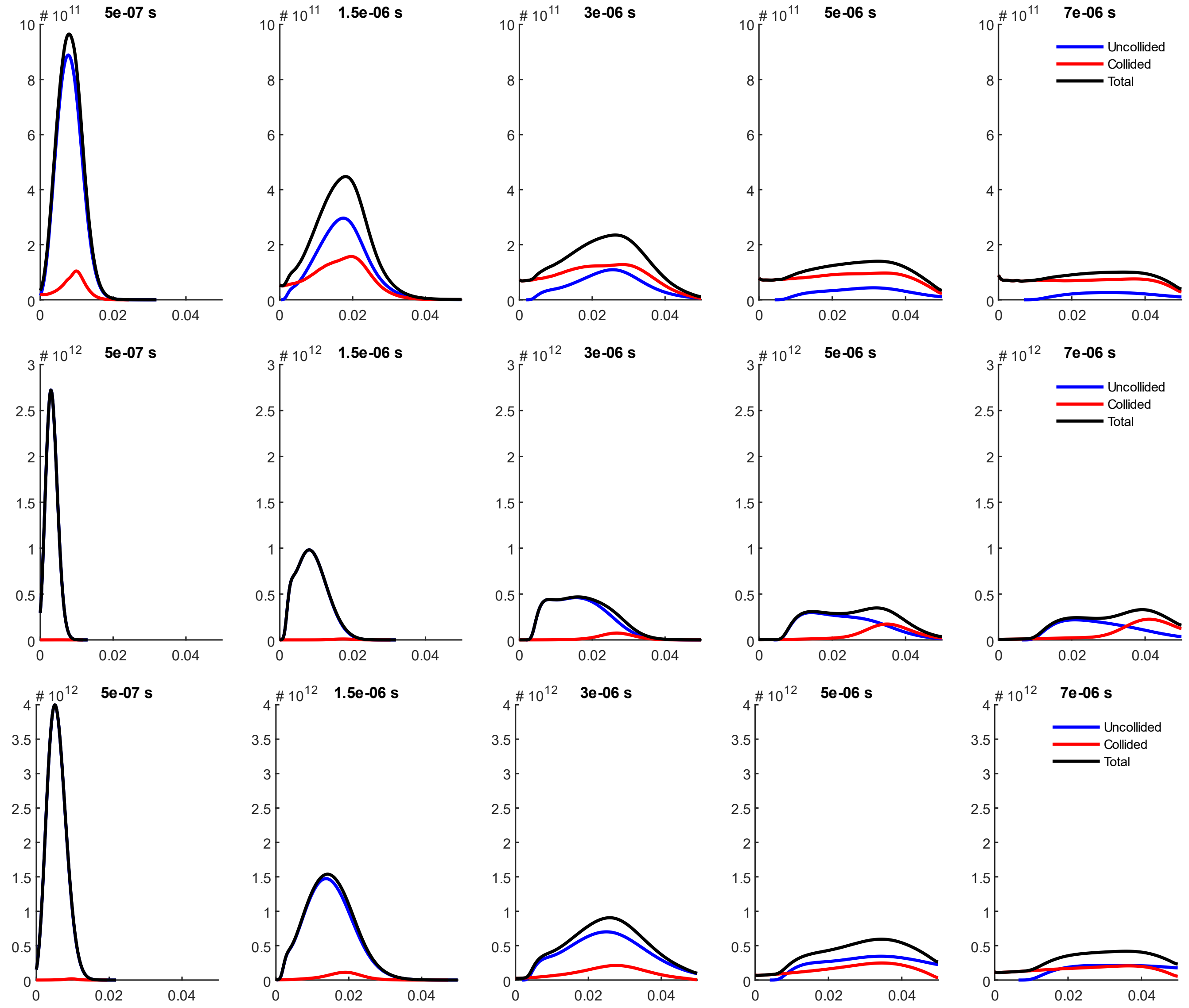



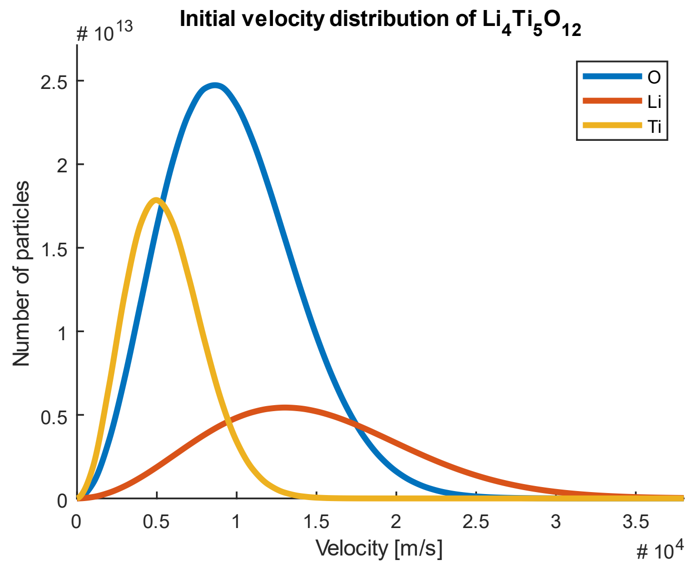
# **Results** Plasma propagation – Li4Ti5O12 at 0.02 mbar

Li

Ti

O



# Conclusions

- The model does not exactly match measurement, but results are in a similar range.
- The model delivers consistent results for different type of materials.
- There seems to be an error for particle moving with a negative velocity, visible at higher pressures

**Improvements**

- Fully modeled background gas particle kinematics
- Support of any target composition, even mixture targets
- Conservation of number of particles is ensured
- Code is dynamic, self-documenting, and contains no ‘magic’ variables

**Features missing**

- Oxidation of plasma species (although data structure supports it)
- 2D plume expansion plots

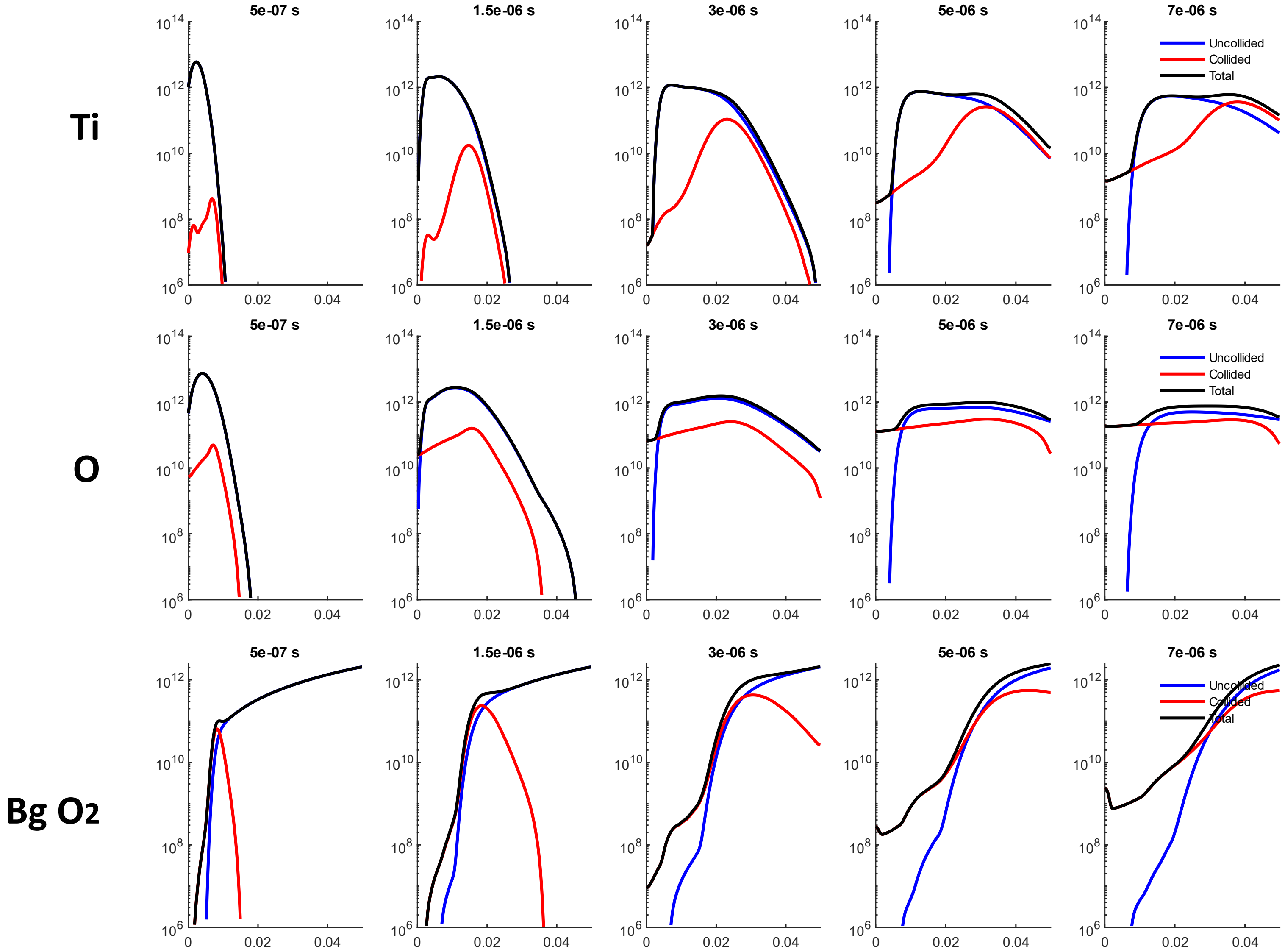
# Suggestions

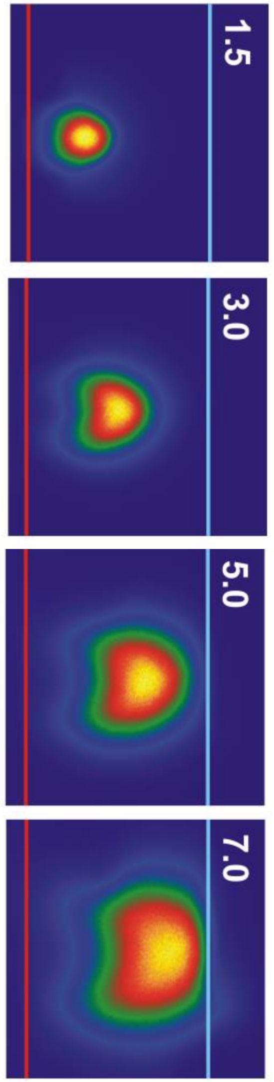
- Introduce a temperature gradient in the background gas based on the substrate temperature
- Find approximations for the heat dissipation into target and excitation energy of atoms
- Include relation between laser fluence and spot dimensions and the angular distribution of particles
- Allow highly kinetic particles to reflect of substrate surface and change direction
- Implement MaterialsProject API to get material properties straight from their database
- Add as material property which oxides a specific atom can form
- **Implement collisions between plasma species !**

# Proposal for collisions between all particles model

- Loop through particle speeds
  - Loop through particle directions (towards substrate or towards target)
    - Loop through plasma species (and additional oxidized species)
      - Loop through filled radial bins
        - Calculate projected path the particle would travel without collisions
        - Calculate the number of collisions with every species and for all radial bins in the projected path (for all slower moving particles) simultaneously using matrix calculations
        - Calculate new velocities and positions after collision
  - Limit number of collisions by the total number of particles available
  - Remove particles from velocity and position before collision
- Update all non-collided particle positions
- Add back the collided particles to their new velocity and position after collision

# Results Plasma propagation – TiO2 at 0.02 mbar (log)



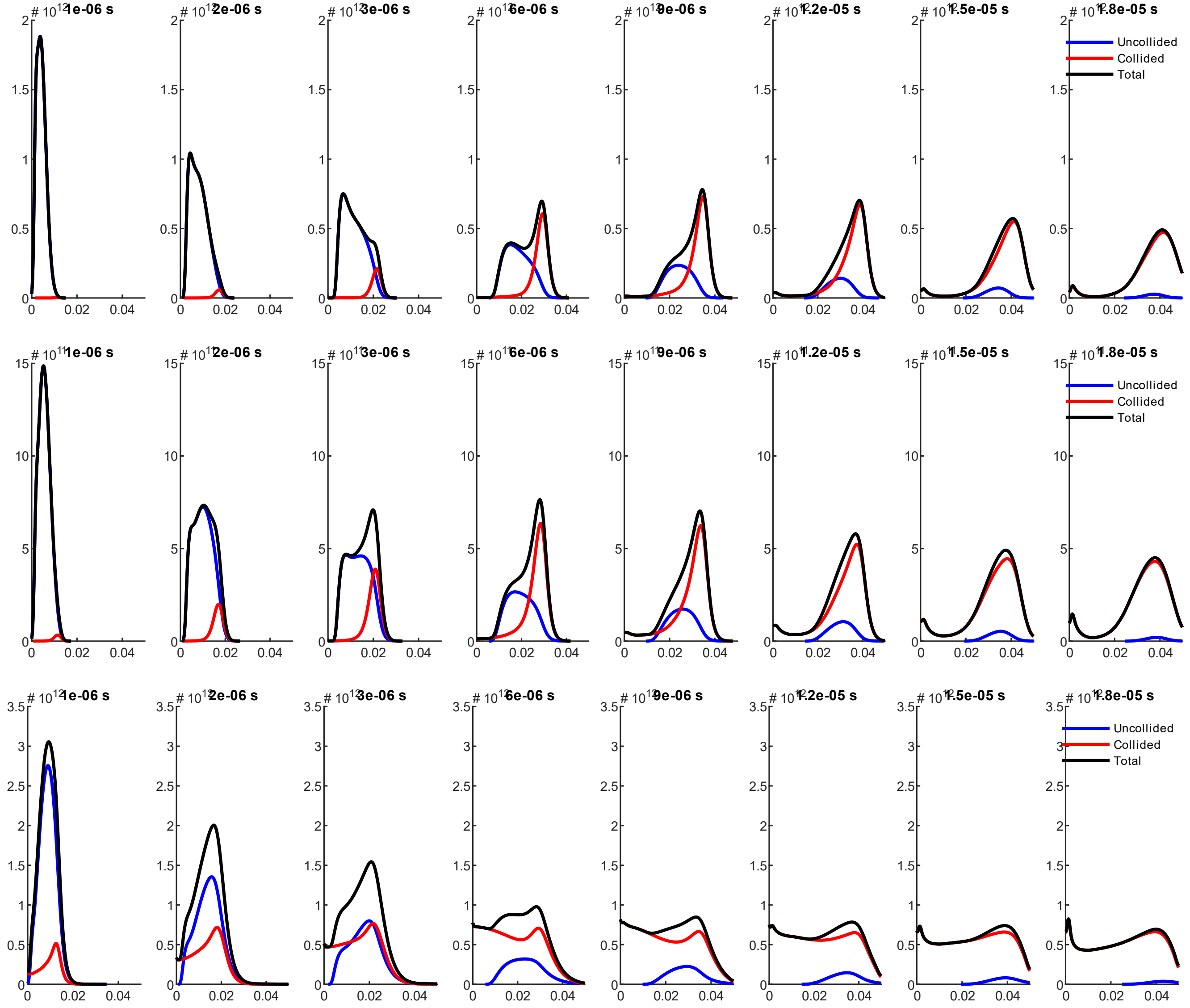



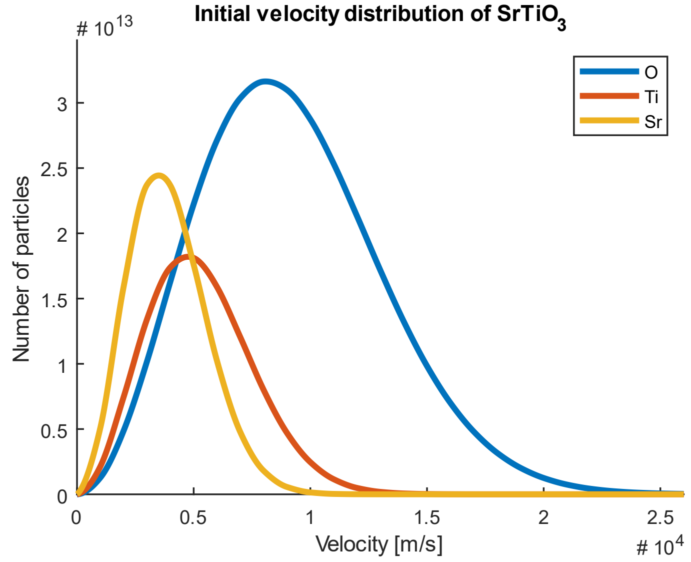
# **Results** Plasma propagation – SrTiO3 at 0.1 mbar

Sr

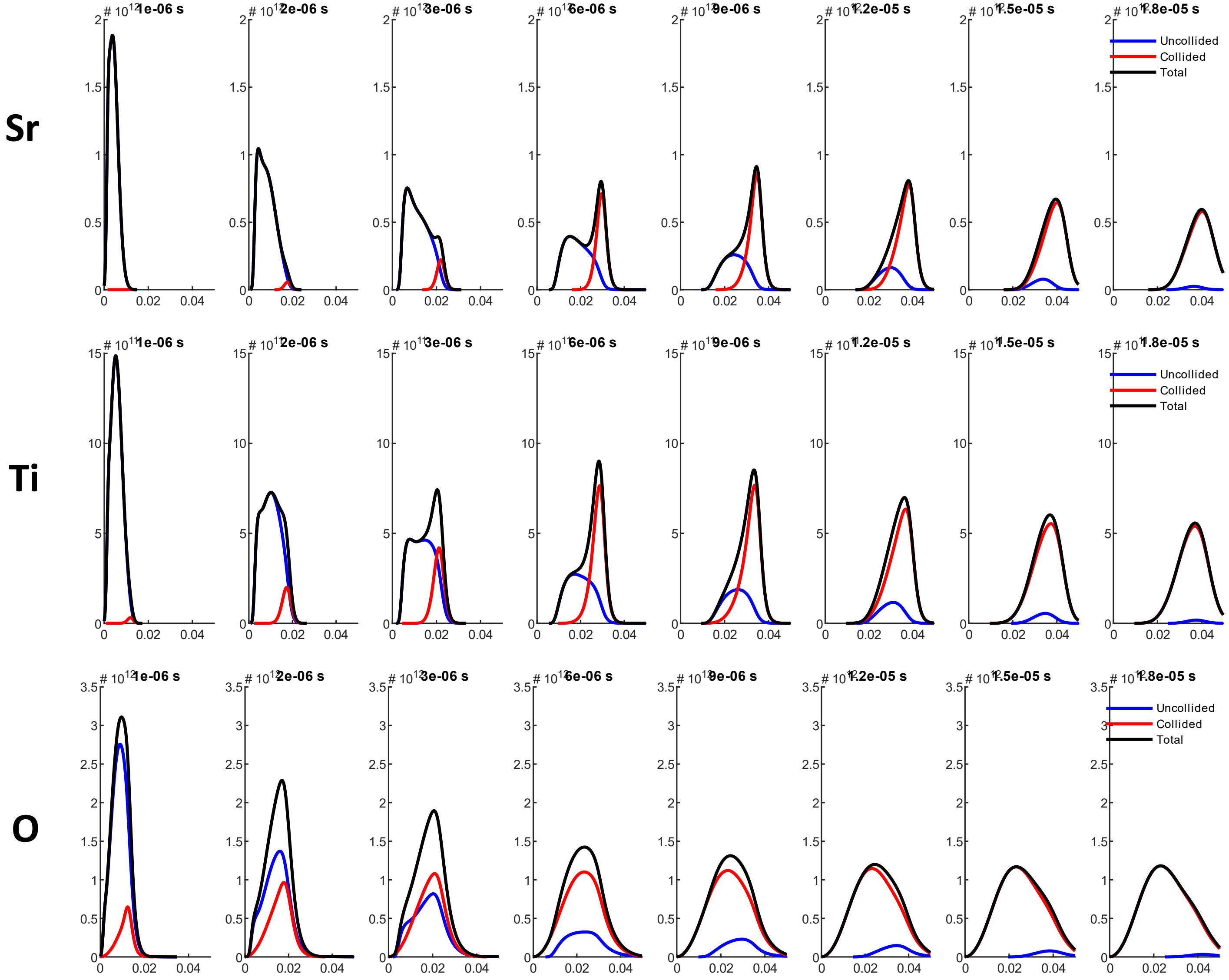
Ti

O

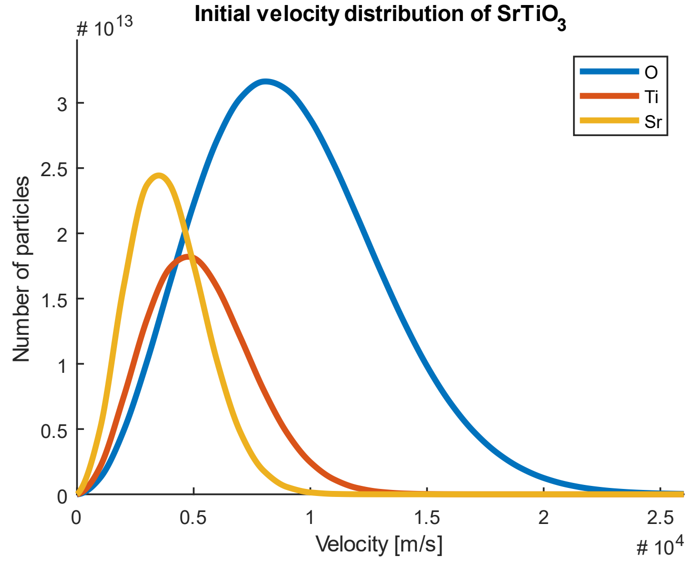


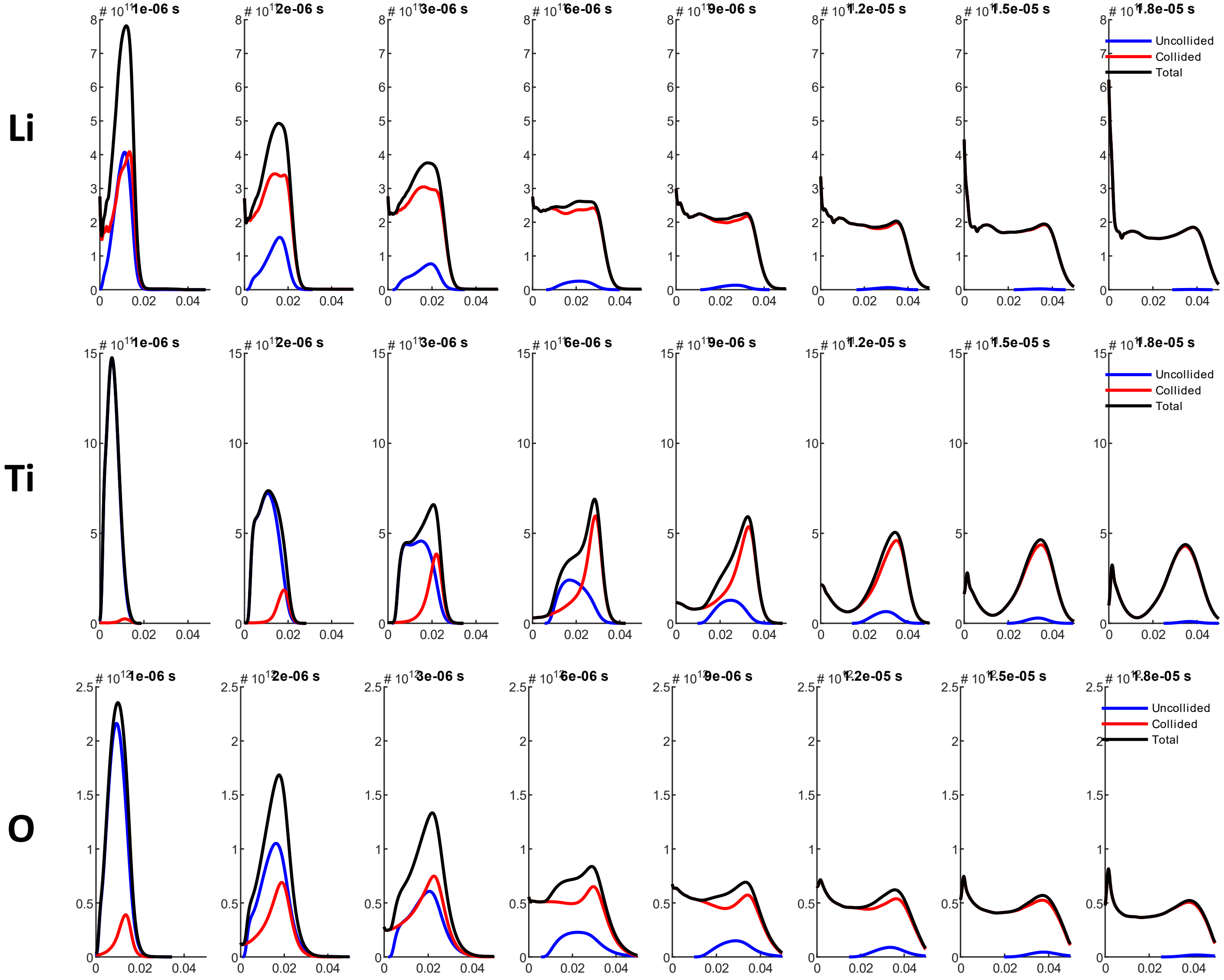
# **Results** Plasma propagation – SrTiO3 at 0.1 mbar

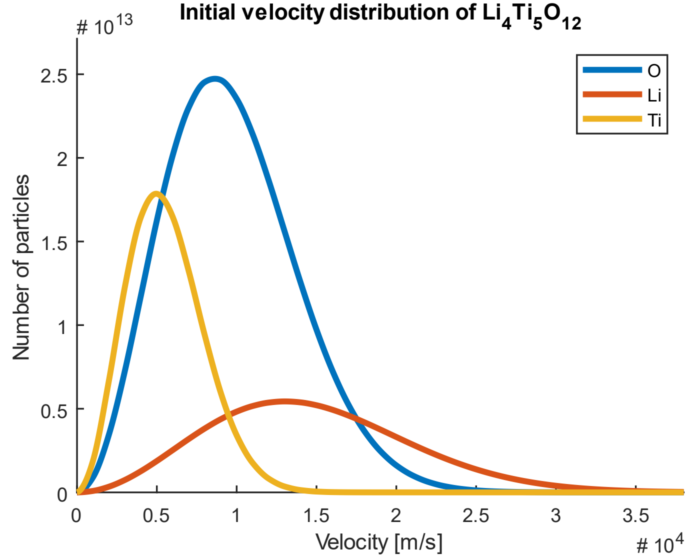



**No negative velocities allowed**

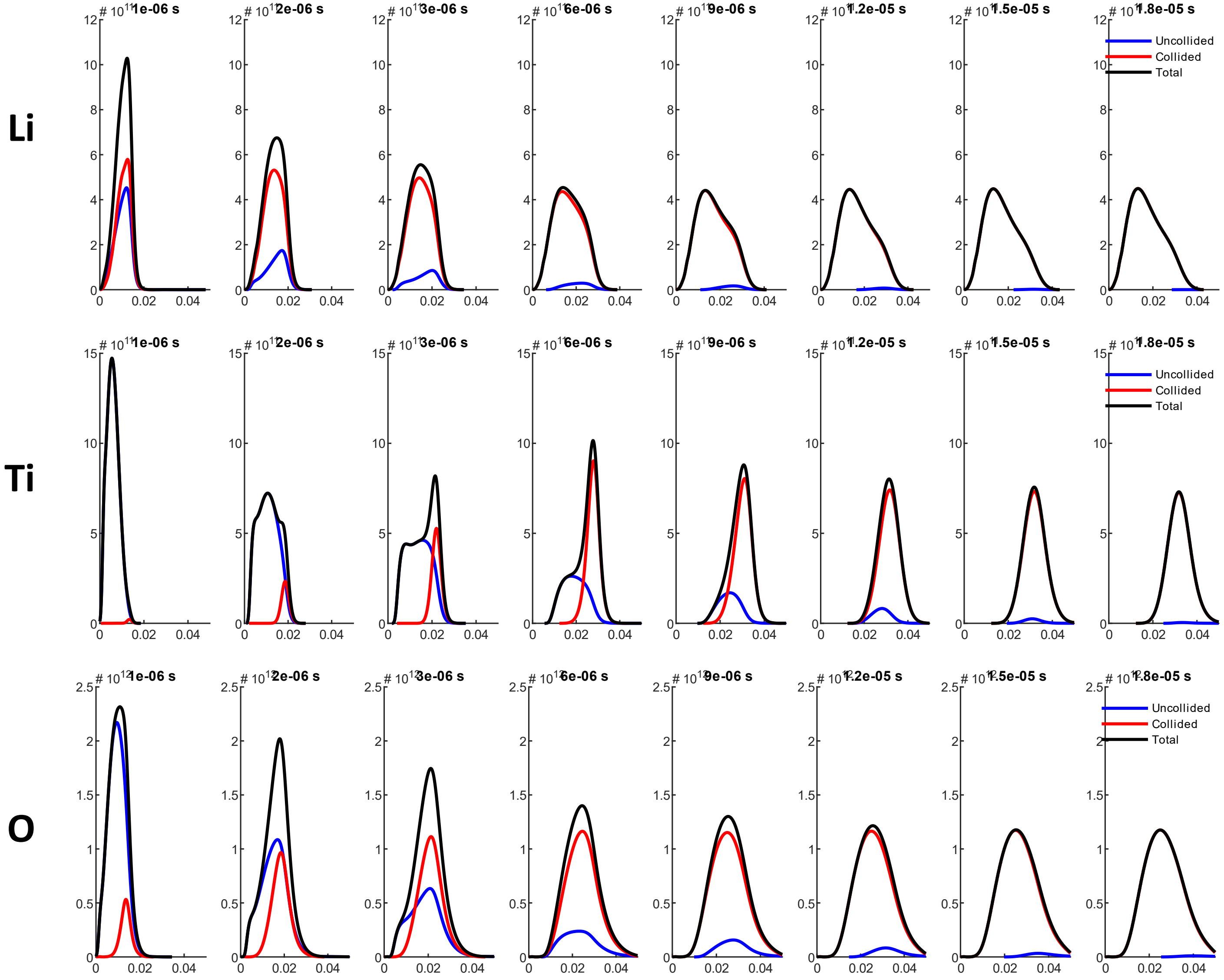


# **Results** Plasma propagation – Li4Ti5O12 at 0.1 mbar



# **Results** Plasma propagation – Li4Ti5O12 at 0.1 mbar




**No negative velocities allowed**