Electron cooling in a cometary coma

Simon Alinder
Abstract

Electron cooling in a cometary coma

Simon Alinder

The ESA Rosetta spacecraft investigated comet 67P/Churyumov-Gerasimenko during two years from August 2014 to the end of September 2016. The dual Langmuir probe was used to measure plasma parameters including the thermal energy of the electrons. The observed thermal energy (or temperature) of the electrons was rather high, in the range 5-10 eV almost throughout the mission. However, near perihelion the Langmuir probe measurements indicated the prevalence of two electron populations with distinct temperatures, one hot (5-10 eV) and one cold (less than 1 eV). It has been hypothesized that the electrons of the colder population were formed relatively close to the nucleus and that they subsequently cooled by inelastic collisions with the neutral gas. In this project work we develop a model for studying electron cooling in a cometary coma. The model takes into account collisions with water molecules as well as the influence of a radial ambipolar electric field.
Sammanfattning

**Contents**

1. Introduction  
2. Background  
3. Theory and Method  
4. Results  
5. Discussion  
6. Recommendations  
7. Conclusions  
8. Acknowledgements  
9. References
1 Introduction

The ESA Rosetta spacecraft resided in the vicinity (typically tens to a few hundred km) of the nucleus of comet 67P/Churyumov-Gerasimenko from August 2014 until September 2016, accompanying the comet through its perihelion passage in August 2015. The coma of 67P consists mostly of water (Hansen et al. 2016). As molecules in the coma are subjected to photoionization, electrons are released with a typical energy of $\sim$10 eV (Vigren & Galand, 2013). In this paper the terms "temperature" and "energy" are used interchangeably. The term "speed" could also be used to mean the same thing, as it is the movement of particles that are referred to in all cases.

The Rosetta spacecraft was equipped with several instruments for measuring properties of the plasma. These measurements showed that the electron temperature at the spacecraft location typically was rather high, in the range of 5-10 eV almost throughout the mission (Odelstad et al. 2015, Eriksson et al. 2017). Near perihelion, however, a population of electrons at the sub eV level was also detected (Eriksson et al., 2017).

At large heliocentric distances ($\sim$3 AU) there were observations of electrons with significantly higher energies ($\sim$100 eV) that even competed with EUV photons as the main ionization source. Various origins of these high energetic electrons have been proposed including energization of photoelectrons by plasma waves and acceleration of solar wind electrons by an ambipolar electric field (Broiles et al. 2016, Madanian et al. 2016).

This project is a modeling study that aims to answer the questions of the origins of these electrons. From our model studies we find that if the electric field is disregarded, the model does not reproduce the observed electron energies. If collisions are disregarded there is no mechanism for changing the path and total energy (potential plus kinetic energy) of the electrons. Only if both collisions and the ambipolar electric field is included does a result emerge, that while not matching the observations, at least resembles them.

Although the model ultimately is shown to not be complete, valuable insights into the mechanics of the electrons in the environment around the nucleus are gained. In section 2 some background is presented, in section 3 some theory is presented along with details of the physics involved and some derivations of used formulas, as well as a presentation of the code, with the complete code included in Appendix A. As is shown in section 4 and discussed in section 5 we can claim that: i) electrons that are registered as cold are not necessary created close to the nucleus, ii) that the ambipolar electric field is very important in the electron cooling process and iii) that collisions are not solely responsible for electron cooling. The ambipolar electric field can serve both to add and remove kinetic energy from the electrons depending on the direction they are traveling in.

2 Background

In this section we will discuss the inception of this project, relevant instruments on the Rosetta spacecraft, some relevant physics and a brief overview of earlier, similar, projects. The Rosetta spacecraft was equipped with two Langmuir probes (LAP), an instrument that can determine electrical properties of
a plasma. By varying the bias voltage a probe collects either ions or electrons. Parameters of the plasma such as the electron number density and the electron temperature can be derived from the current-voltage characteristics. The probes can operate by doing “sweeps” by varying the voltage on the probes and analysing the current generated by particles from the plasma being attracted to the probes. Using both probes allows for estimation of the electric field from the voltage difference and separation of the probes (Eriksson et al. 2007; Edberg et al. 2015; Karlsson et al. 2017). These measurements by LAP showed that the electron temperature at the spacecraft location typically was rather high, in the range of 5-10 eV throughout the mission. Near perihelion, however, the bias voltage sweeps often required consideration of two electron populations for proper fitting of the data. Apart from the hot population, a population of cold electrons at the sub eV level was also seen (Eriksson et al. 2017).

A solar wind void around 67P was revealed by the absence of solar wind particles at the spacecraft location between April and December 2015 (Nilsson et al., 2017, under review). This follows from mass loading of the solar wind. Cometary ions are picked up and accelerated by the convective electric field of the solar wind and the solar wind ions are deflected in opposite direction (momentum conservation) ultimately yielding a void in the vicinity of the nucleus. This means that the environment close to the nucleus, inside the void, is different and largely protected against the effects of the solar wind. It is therefore suitable to use this void as the region for the simulation.

A current idea is that the observed cold electrons originated closer to the nucleus, and that they were efficiently cooled by inelastic collisions with molecules in the coma, before reaching the spacecraft location. Simple analytic models of cometary ionospheres have been put forward wherein electrons produced by photoionization within a certain distance $r_{en}$ are assumed cold (sub eV) while those produced outside $r_{en}$ are assumed hot (5-10 eV) (Mandt et al. 2016, Eriksson et al. 2017). One aim of the present project is to test the feasibility of such a model.

This project consisted of the creation of a computer model to simulate the energies and paths of electrons in the vicinity of a comet nucleus. By keeping track of how much energy is lost through interactions with the neutrals and how much is gained or lost by interacting with the electric field we can trace the origins of particles at the end of the simulation.

A model for electron cooling was put fourth by Cravens (1986). They presented energy loss functions for the electrons by computing quantum mechanical interactions between electrons and water molecules including rotational and vibrational excitation. One shortcoming of their method is that the electrons implicitly are assumed to be in the same environment during their whole considered lifetime, something that is not necessary true as the electrons move with very high velocity and the environment changes with distance from the nucleus. This model was developed with the comet 1P/Halley in mind, a much more active comet compared to 67P/CG, so the model presented in the paper is more valid for that environment as a more active nucleus means that all regions and features around it are bigger.

A far more complex model can be created using MHD-, hybrid- or Particle-In-Cell simulations (e.g., Koeenders et al. 2014, Rubín et al., 2014; Deca et al. 2017). However, our main focus is on the energy evolution and trajectory of electrons (test particles) in a preset radial electric field. Besides, this project
has to fit inside the scope of a 15 credits course.

This project was a simulation of the behavior of electrons close to the comet nucleus, inside a solar wind void and diamagnetic cavity. The main purpose of this simulation is to evaluate the feasibility of a simple model wherein all electrons created within a distance $r_{en}$ are cold and all electrons created outside this distance are warm.

3 Theory and Method

In this section we will present the physics the code tried to simulate and how it was done. Points we go through are: the general approach, general physics, electromagnetism around the nucleus, types of interaction between electrons and H$_2$O molecules and how the code works.

This project was a modeling study aimed at estimating the expected energy distribution of photoelectrons at some distance from the comet nucleus. The general approach to this problem was that of making an intuitive and deep simulation that can answer many different questions, even ones not foreseen in the planning of the project. For this reason a full, kinetic, 3D model was chosen as the basis of the simulation. This approach allows us to easily plot the trajectories of particles and debugging was made easier when particle behavior could be directly observed. The software used was MathWorks’ MATLAB.

A spherically symmetric coma was assumed with a number density of H$_2$O molecules that decays with cometocentric distance $r$ according to:

$$n_N = \frac{Q}{4\pi vr^2},$$

where $Q$ is the outgassing rate of the nucleus, set to $2 \times 10^{28} \text{s}^{-1}$ and $v$ is the velocity of the water molecules, set to $1000 \text{ m/s}$ to simulate conditions at perihelion (e.g., Hansen et al., 2016). The probability of collision was quantified from the ambient H$_2$O number density and the total collision cross-section which is energy dependent, taken from the review by Itikawa & Mason (2005). The values of the cross-sections for various processes as well as total cross-section was also taken from the review. A simple table look-up script with basic interpolation was used for each one and the values were pulled from there.

At high activity the nucleus is surrounded by a solar wind void and a diamagnetic cavity, later observed intermittently and particularly frequently near perihelion (Goetz et al., 2016). This means that the effects of the solar wind are not strongly felt close to the nucleus, and can be neglected. The water plasma in the coma is quasi-neutral but, when a water molecule is ionized, the free electron moves at a velocity that is orders of magnitude higher than the ion. This causes a separation of charges that creates an electric field, called an ambipolar field. The ambipolar field is just the electric field created by the separation of charges in the plasma.

We consider only a radial electric field set up by the electron pressure gradient force

$$E_r = -\frac{\nabla p_e}{n_e q},$$

(2)
where $p_e$ is the density of electrons, calculated as

$$p_e = n_e k_B T_e$$

in SI units where $k_B$ is Boltzmann’s constant and $n_e$ is the number density of electrons. A default ambipolar electric field is given by:

$$E_{amb} = T \frac{r - 2r_c}{r(r - r_c)}$$

where $T$ is the average electron temperature in eV and $r_c$ is the radius of the comet nucleus, which is assumed to be 2 km. $T$ is assumed to be 10 eV based on measurements from the Rosetta spacecraft (see section 1). Equation 4 follows from 2 and 3 when assuming $n_e \propto \frac{r - r_c}{r^2}$ and a constant $T$. A rough $1/r$ relation was observed for the electron number density during a radial scan at low activity (Edberg et al., 2015) while changes in the electron temperature were smaller. The utilized default electric field is based on the assumption of collisionally coupled ions (moving radially outwards with the same speed as the neutrals) yielding an ion density profile (Galand et al. 2016) proportional to $\frac{1}{r^2}$. This also means that the density of ions is proportional to the density of neutrals.

To simplify the simulation, any magnetic field is neglected. The size of the region considered is within 100 to 200 km from the comet nucleus, where frequent diamagnetic crossings were observed. We assume that at the edge of the diamagnetic cavity a pileup of magnetic field lines prevent particles outside from entering, this aligns with the model not creating particles outside the region considered.

The nucleus also has a non-zero electric potential. This is because the electrons have much greater speed than the ions. So if the nucleus is at zero potential, it is hit by many more electrons than positive ions, and so gets negatively charged until the fluxes of ions and electrons are equal. This typically means charging to a voltage a few times the electron energy (in eV, if all electrons are cold, this effect is small) and therefore also generates an electric field. However, ions in the plasma get attracted to the nucleus and form a shell around it, canceling out most of the field if seen from a distance. The charge of the nucleus was simulated by having particles of low energy bounce of it and particles with high energy colliding with it. This was meant to simulate the short range repulsion which the low energy particles could not overcome, but the high energy ones could.

The code was meant to simulate the trajectory of a single particle at a time. Initial energies were chosen according to a solar photo electron spectrum (see figure 22). Starting locations and directions were randomised within set boundaries. The model was a stochastic simulation in which a particle and its trajectory was determined by kinematics and stochastic interactions. The path of the particle changes in every point of its discretized trajectory, because it is accelerated by an electric field described in equation 4. This simulation was based on determining the electric force acting on the particle at every step and changing its direction and velocity accordingly, as well as calculating the probability of interaction in each step and doing an interaction if it happens.

The probability of interaction $P$ within the length of the step $ds$ was computed using a Beer-Lambert type equation:

$$P(E, r) = 1 - \exp \left( -\sigma(E) \ast n_N(r) \ast ds \right)$$
where $\sigma(E)$ is the cross-section of an electron scattering on a H$_2$O molecule (from Itikawa & Mason (2005)) and $n_N$ is the density of neutral particles (H$_2$O molecules) seen in equation 1.

When a particle interacts with a target, the type of interaction is determined, based on a comparison between the total cross-section and the cross-section of the specific interaction (both taken from Itikawa & Mason (2005)). The interactions considered in this project were elastic collisions and inelastic collisions including ionizations and electronic excitation (including pathways leading to dissociation). Elastic collisions and rotational excitation were combined, in these interactions the impinging electron loses very little energy, on average only 4 meV. In an inelastic collision with vibrational excitation the electron transfers some energy to the water molecule by exciting it either by bending or stretching the molecule, with energy losses of 0.198 eV and 0.453 eV, respectively. Electronic excitation typically gives an electron energy loss in the range of 7-11 eV, including pathways leading to emission as well as dissociation. In ionization events the electron loses at least 12.6 eV of energy, the lowest ionization potential of H$_2$O. The scattering angle for inelastic collisions with vibrational excitation were set to be a fixed value of 5 degrees, chosen arbitrarily to respect the strong tendency for forward-scattering i.e. that electrons usually continues along a similar trajectory after interaction. For every other type of interaction, a random angle is chosen for the particle to scatter along, with a heavy bias towards low angles.

When the mechanics of the collision has been determined, the direction of motion is rotated according to the angle found in the interaction. Then the loss of energy is applied, and the next step is taken. If the particle does not interact during the current time step, its direction is rotated according to the acceleration from the electric field and its position is updated. If the particle loses almost all of its energy (down to the order of a few meV) the particle is kept at a positive, non-zero energy, unless it gains energy by being accelerated by the electric field. The exception is if the particle is in the region where the E-field changes sign. In that case the particle will become trapped and essentially vibrating between two points, with no mechanism for exiting the region and is therefore removed from the simulation. The simulation then continues with the next step, repeating until any of the conditions for stopping are met.

The simulation uses a variable time step which is determined by the density of neutrals modeled by the distance to the nucleus. A special case is applied if the particle is very close to the nucleus or heading for a collision with the nucleus. In these cases a shorter time step is used.

The conditions for stopping are:

1. The particle reaches a set distance from the nucleus and escapes the region (100-200 km is default).
2. The particle gets to close to the nucleus and is assumed to collide is its energy is high enough (a homogeneous radius of 2 km is assumed and energy of more than 30 eV is required).
3. The energy of the particle falls below a threshold and is within 100 m of the distance where the E-field changes sign (1 meV per default and the field changes at 2 $r_c$ m from the origin).
4. The cumulative time step reaches some set limit (1 s is default).
The ratio of hot to cold particles is computed at the end of the simulation, and an energy spectrum is created to view the general results, as well as several other figures presented in section 4.

The simulation uses a system of concentric "shells", spaced 1 km apart, to register the energies of the particles. Each time a particle passes through a shell its energy and angle to the shell is recorded. The latter is used to compute the mean energy as a function of distance as well as an indicator of the electron density.

4 Results

In this section, figures and data generated by the code are presented. Figure 1 shows the mean energy of particles as a function of conoentric distance. Figure 2 shows the "density indicator" (a measure of the flow through an area in space) as function of conoentric distance. This is not the same as the actual density of the particles but it should have the same shape. Figure 3 shows a comparison between the theoretical ambipolar E-field determined by equation 4 and an E-field computed from the motion of the particles in the simulation. Figures 4, 5 and 6 shows amplitude spectra of the computed E-fields like the one in figure 1 for different amounts of particles.

Figure 7 shows the energy distribution of all particles that passed through the "shell" at 99.7 km from the center of the nucleus. The particles were not affected by the passage and some passed the shell more than once. This happens when any of the mechanisms for changing the direction of particles causes them to do so, see section 3. For this reason there are more registered passages than particles in the simulation. Figures 8 and 9 are arrays of histograms like the one in figure 7. Figure 10 is a histogram showing the energy each particle had when it was removed. This means every particle is counted precisely once. Figures 11 and 12 show the positions of particles that became cold at the start and end of the simulation. Figures 13 and 14 show the probability for an electron to end up cold or escaping the region respectively. These graphs were generated by using the shell structure in the model and counting how many particles were created between each pair of shells, then computing the percentage of particles that ended up escaping or becoming cold. Figures 15 and 16 show examples of paths of single particles in the simulation. The color of the line indicates the energy of the particle.

Figures 17 through 21 are final energy histograms like figure 10 but with some special setting. Figure 17 is the result of a run with no electric field, figure 18 is the result of a run in which all particles start by moving radially outward, figure 19 is the result of a run in which all particles start by moving radially inward, figure 20 and 21 are the results of runs in which all particles start at a certain distance from the nucleus, 10 km and 100 km respectively, with random starting energies and directions. Table 1 shows how big a part of the particles ended up as escaped, warm or cold in simulations with the regular E-field and without any E-field. Figure 22 is a histogram showing the energy distribution of 1 million particles created according to the photoelectron spectrum.

The number of particles used in the simulation that yielded the figure is usually 10000, and is noted in the caption. 10000 was chosen as it is a high number which suppresses stochastic variations but so large as to make the run-
time of the simulations to long. The histograms (particularly figures 8 and 9) have noted the number of registered particles in the figure. This number is generally not the same as the number of particles in the simulation for the reasons explained in the second paragraph in this section.

<table>
<thead>
<tr>
<th>Fate</th>
<th>Default E-field</th>
<th>No E-field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Escaped</td>
<td>45.7%</td>
<td>77.5%</td>
</tr>
<tr>
<td>Stayed, warm</td>
<td>38.3%</td>
<td>2.5%</td>
</tr>
<tr>
<td>Stayed, cold</td>
<td>16.0%</td>
<td>20%</td>
</tr>
</tbody>
</table>

Table 1: Fates of particles
Figure 1: Mean energy of particles as a function of radial distance, 10000 particles
Figure 2: Density indicator as a function of radial distance, 10000 particles
Figure 3: E-field, 10000 particles
Figure 4: Fourier spectrum of calculated E-field, 10000 particles

Figure 5: Fourier spectrum of calculated E-field, 5000 particles
Figure 6: Fourier spectrum of calculated E-field, 2500 particles
Figure 7: Energy distribution at 100 km from nucleus, 10000 particles
Figure 8: Energy distributions at 15 km intervals, 10000 particles

Figure 9: Energy distributions at 1 km intervals, starting as close as possible as the radius of the nucleus is 2 km, 10000 particles
Figure 10: Final energy distribution, 10000 particles
Figure 11: Where the particles that were cold ended up and where they were created, 10000 particles.
Figure 12: Detail of top figure in figure 11
Figure 13: Probability that particle becomes cold as a function of how far from nucleus it is created, 10000 particles
Figure 14: Probability that particle escapes as a function of how far from nucleus it is created, 10000 particles.
Figure 15: Path of a single particle that timed out. Comet nucleus at origin marked by circle. Color of line indicates energy. Energy varies between 6.5 and 0 eV.
Figure 16: Simple path of a single particle that escapes. Comet nucleus at origin marked by circle. Color of line indicates energy. Energy varies between 23 and 17 eV.
Figure 17: Final energy distribution with no electric field, 10000 particles
Figure 18: Final energy distribution, all particles starting radial
Figure 19: Final energy distribution, all particles starting anti-radial
Figure 20: Final energy spectrum of 5000 particles starting at 10 km
Figure 21: Final energy spectrum of 5000 particles starting at 100 km
Figure 22: Histogram showing the energy distribution of 1 million particles created according to the photoelectron spectrum. X-axis is energy in eV and y-axis is number of particles created at that energy. Source: E. Vigren and M. Galand (2013)
5 Discussion

All values for cross-sections for all processes are taken from Itikawa & Mason (2005) and all uncertainties that are discussed in that paper carry over to this one. The default electric field is assumed to be a smooth function of cometocentric distance and it is stressed that the simulations are intended mainly for distances within the diamagnetic cavity. Observations outside the cavity reveal a highly dynamic environment with wave activity and variable (over short time scales) electric and magnetic fields as well as variable electron number densities, indicative of a highly dynamic plasma (e.g., Edberg et al., 2015, Karlsson et al., 2017, Henri et al., 2017, under review).

The time scales that the code operate on represents a fraction of a second for the particles that escape and a maximum of a few seconds for those who do not. For this reason the movement of the ions is neglected as they will move a very small part of the 100-200 km distance considered in the simulation.

As stated at the end of section 3, the code uses shell-crossings to count the particles. This approach has a flaw, which is that under certain circumstances a particle may stay between two neighboring shells. If the particle has a very low energy it may still move around but change direction very often and use very short steps. This means that the information of this particle is not updated, and cold particles may be underrepresented in some of the results. This being said, even if the particles have very low energies the electric field does move them unless they are in the region where the E-field changes sign. Looking at figure 2 it appears that plenty of particles are being registered even in the region where the collisional cooling is the strongest and the mean energy is low according to figure 1.

An updated electric field profile was derived from cataloging the test particles, using the mean energy of particles crossing a shell as a temperature estimate (see figure 1) and the sum of all individual crossings density contributions (an arbitrary number divided by the speed in the radial direction) as an electron number density indicator (see figure 2). The values for different cometocentric distances were inserted into equation 2 and $E_r$ was calculated numerically. The result is displayed in figure 3. The calculated $E_r$ was found to smoothen with an increased amount of test particles (see figures 4, 5, and 6) and noise in the profile are thus largely attributed to the stochastic nature of the model. When iterating the electric field (i.e. using the calculated electric field from run N in run N+1 a couple of times) we identified a self-amplifying situation with ever greater values of $E_r$ for a given cometocentric distance.

This reveals shortcomings of our approach and perhaps missing physics. For one it is not obvious how to treat long lived particles trapped in orbits in the simulation box. The number of shell crossings such a particle makes, and thus the mean energy and in particular the density indicator, will depend on the simulation time utilized. The possible removal of such particles through dissociative recombination with molecular ions must be considered in an updated version of the model. Coulomb interactions (ion-electron as well as electron-electron) may also be important to account for. Interactions between electrons with electrons or electrons with ions are expected the most efficient in high density regions (closer to the nucleus) which thus can be expected to reduce the electron number density gradient. Finally, our treatment of the electron pressure is questionable; is it strictly correct to approximate the electron temperature
by a mean energy? Interestingly, Madanian et al. (2016) also found that the consideration of the ambipolar electric field in their two-stream model brought an unstable solution.

One potential source of these errors is how the data is handled. Information of the motion of the particles is extracted from the 3 dimensional model and decomposed into a 1 dimensional vector. This vector is then used to impose a new E-field. This means that any local variations are applied to the whole region, a randomly created pocket of higher density becomes a full shell and thus overstating its effects.

The simulation can be run with different electrical fields. As seen in table 1 (and figures 10 and 17), if the model does not include any electric field, the particles will mostly escape and those that do not are cold (and usually trapped without any way of escaping). This suggests that the ambipolar field is an important component in the electron cooling mechanism. However, our simulation fails to explain how significant amounts of sub-eV electrons can reach the spacecraft location, at a distance 200 km from the nucleus.

As seen in figure 7, the simulation does not reproduce the observed shape of the energy distribution, with two peaks, one for high energy electrons and one for low energy ones. This could be due to a flaw in the model or some more complex physics that cause the observed phenomenon in reality, that is not included in the model. The ambipolar field helps to "cool" (reduce the kinetic energy of) outward moving particles, and the very low-energy particles are typically brought inwards before reaching the edge of the region. The vague two-peak like structure in the bottom right image in figure 5 is a result of the starting energy of the particles following the photoelectron spectrum seen in figure 22. Many of the particles created in the >25 eV region would make it out to 150 km with high energy and show up in the figure.

The starting and stopping positions of particles that become cold can also be tracked, and by looking at figure 11 we see that particles that become cold may start anywhere (although more commonly close to the nucleus), but they almost always end up close to the nucleus. The boundary of the region where the most cold particles end up is not sharp, but as seen in figure 12 if a radial distance of around 30 km is chosen it will include almost all particles and not a lot of empty space. This results does not support the assumption that electrons produced at distance from the nucleus remains hot. However, as expected, the most efficient cooling occurs close to the nucleus.

In figures 13 and 14 we can see the probability that a particle escapes or end up becoming cold as a function of starting distance from the nucleus. There is no clear limit where the probability for an particle to become cold jumps up, but we can see that the probability is low outside of around 40 km and rising quickly inside of that. We can read of the point where it becomes more likely for an particle to end up cold than not, and that point is around 15 km from the nucleus. We see that the probability for a particle to escape goes up approximately linearly, and is zero for the first couple of km. Both graphs show that the probability of becoming cold or escaping becomes very high at the respective edges of the graphs, which is to be expected.

During four runs of the simulation special restrictions where imposed on the starting situation of the particles. For the first two the starting direction of all particles was restricted to being radial and anti-radial respectively. The simulations had the same parameters in every other way. The resulting final
energy distribution can be seen in figures 18 and 19. As could be expected, when all particles start going out from the comet, more of them escape the region after a shorter time which results in higher retention of energy for those particles. Many more particles ended up in the lowest energy bin when all particles started by going inward, indicating that particles lose more energy by moving closer to the nucleus.

The later two starts had all particles starting at the same cometocentric distance, 10 and 100 km respectively. Looking at figure 20, we see that a very large part of the particles finish the simulation with little to no energy left. When comparing this to figure 21, we can argue that a simple model may be acceptable. With around 75% of particles starting 10 km from the nucleus ending up with <1 eV but less than 20% doing the same when they start 100 km from the nucleus. No particles with energies above 30 eV are seen in figures 7 and 10. This indicates that, even though the particles can gain energy in excess of their starting energy in this model, the maximum energies reached are lower than the observed energies for the “mysterious high energy electrons” with energies of ~100 eV. It is possible that at weaker activity and well outside the diamagnetic cavity electrons can be accelerated over much longer distances and by stronger fields than those considered in our simulation.

The shape of the curve in figure 1 can be understood by considering the following. Far from the nucleus, at distances greater than 125 km in this case, only the particles that were created with high energies remain, the others have been pulled in towards the nucleus by the electric field. And most of the particles that are created with high energies are likely to make it out to this region unless they appear very close to the nucleus. This means that the average energy in this region is high. Very close to the nucleus, inside of ~10 km in this simulation, the collisions will cool the electrons and effectively lower the average energy. In the middle of the region is where particles with low energies that are moving outwards turn back, and therefore have very low energies. Between the last two regions is the peak at around 20 km, this is formed by electrons coming in from the further regions, accelerated by the electric field to higher energies but not yet heavily cooled by collision.

A slightly increasing trend can be seen at the end of the generated E-field in figure 4. This is due to the region being close to the edge of the region considered, and no particles can come in from outside. Therefore almost all particles are moving outwards. This can be somewhat compensated for by focusing on the results closer to the nucleus.

A sensitivity analysis of the code was conducted, in which the values of certain key variables were changed to see the effects in the results. If the results of the code are the same or similar, we can believe that the results were not a coincidence the first time and that the simulation is valid, at least in this limited sense. The variables that were changed were:

- $\hat{Q}$, the activity of the nucleus.
- $\sigma$, the total cross-section of collision.
- $r_C$, the radius of the comet.

The analysis showed the simulation to be very stable, with variations within reasonable margins not affecting our qualitative conclusions.
6 Recommendations

If this work is to be continued, the author would like to recommend some improvements that may be done in order to solve some of the issues that the current model has.

- Use recombination as the mechanism for removal of particles instead of an arbitrary time limit.
- Consider the non-homogeneous angular density of neutrals around the comet, maybe this will provide an important way of transporting cold particles out of the inner regions.

One may also consider expanding the model to include some of the physics that was neglected in this project. In that case the author recommend these additions:

- Include magnetic fields.
- Ion-electron interaction and electron-electron interactions.
- Consider the motion of ions.
- Use a “better” density indicator, however that may look.
- Improve the model to create a self consistent electric field.

7 Conclusions

The simulation of electrons around the comet nucleus shows that the ambipolar electric field has a strong effect on the average electron energy over distance and the ultimate fate of the electrons. For large enough distances the ambipolar electric field decelerates outward moving electrons while accelerating inward moving ones.

The very high energy electrons (>100 eV) observed at low activity and suggested by Madanian et al. (2016) to be caused by solar wind electrons accelerated by the ambipolar electric field, were not seen in our simulation as we restricted the release of electrons to distances <200 km from the nucleus and did not allow for penetration of electrons from outwards into the cavity.

The results from the simulation, if run with a large number of particles to suppress variability, show qualitative agreement with simple models wherein all electrons produced inside a certain cometocentric distance $r_{en}$ become cold, while others tend to remain hot. However, the transition occurs, according to our simulation results, much closer to the nucleus than theorized by others. The value for nucleus activity $Q$ used in this project would give a value for $r_{en}$ of around 80 km according to Mandt et al. (2016). The results from this project does not give a sharp transition, but according to figure 13, the value of $r_{en}$ is around 10-30 km. A remaining key question is also how electrons cooled near the nucleus manage to reach out to the spacecraft location.
8 Acknowledgements

The author would like to extend a thank you to everyone that was helpful, and a special thanks to Erik Vigen for being a good supervisor, Anders Eriksson for some good ideas, Mats André for some valuable comments, William Redlund for giving some good feedback and the whole Rosetta team in Uppsala for their interesting research.

9 References


A. I. Eriksson et al., "Cold and warm electrons at comet 67P". Astronomy and Astrophysics (2016) 301;59; doi:10.1051/0004-6361/201629159

S. A. Fuselier et al., "ROSINA/DFMS and IES observations of 67P: Ion-neutral chemistry in the coma of a weakly outgassing comet". A&A 583, A2 (2015) DOI: 10.1051/0004-6361/201526210


T. Karlsson et al., "Rosetta measurements of lower hybrid frequency range electric field oscillations in the plasma environment of comet 67P". Geophys.


Appendix A

Here follows all the essential MATLAB code used in the project.

**Main code**

```matlab
clear
close all

n = input('Number of particles: ');
start = input('Maximum starting distance for electrons: ');
stop = input('Maximum distance allowed, must be greater or equal to ... starting distance: ');
sat = input('Distance to satellite, must be inside maximum ... distance: ');
cold_limit = 0.5; % Under detta antal eV räknas elektronen som kall
time_limit = 1;
shells = ceil(((max(start,stop)-2e3)/1e3)-2);
targets = linspace(2e3,max(start,stop),shells+2);
targets = targets(2:end-1);
E_kind = choosedialog;
quest = input('Save calculated E-field to use next time? [Y/N]: ','s');
disp(['Using ',num2str(shells),' shells.'])

global i_count
i_count = 0;
if isequal(E_kind,'Custom')
    global t s
    s = load('E_rad_sav.mat');
t = s.targets;
s = s.E_rad;
end
[M,I] = min(abs(targets-sat));
if any(targets==sat) % finns redan
    sat_shell = find(sat==targets);
elseif M > 1e3
    targets = sort([targets sat]); % Sätt in ett extra lager där ... satelliten är
    shells = length(targets); % Så all kod använder rätt antal lager
    sat_shell = find(sat==targets);
else
    sat_shell = I; % Inget nytt skal
end
F = zeros(1,n);
full_E = [];
store_S = [];
shell_En = [];
shell_an = [];
E0_last = [];
E0_first = [];
Esc_first = [];
first_A = zeros(n,3);
```
45 fin_E = zeros(1,n);
46 marker = zeros(1,n+1);
47 marker(1) = 0;
48 E0_count = 0;
49 Esc_count = 0;
50 max_factor = 25;
51
52 for i = 1:n
53 A = rand_A_expand(start); % Start är övre gräns
54 Z = Z_func();
55 E = rand_E;
56
57 [flag, count, store_pos, store_E, e, shell_E, shell_a, ...
58     store_s] = ...
59     newton_sim(A,Z,E,stop,targets,E_kind,time_limit);
60
61 marker(i+1) = count-1;
62 F(i) = flag;
63 a = sum(marker(1:i))+1;
64 b = sum(marker);
65 full_E(a:b) = store_E;
66 store_E(a:b) = store_s;
67 Sx(a:b) = store_pos(:,1);
68 Sy(a:b) = store_pos(:,2);
69 Sz(a:b) = store_pos(:,3);
70 fin_E(i) = e;
71 if size(shell_E,2) < shells
72     shell_E(1,shells)=0; % för att tvinga samma ...
73     dimensionalitet
74     shell_a(1,shells)=0;
75 end
76
77 shell_En = [shell_En;shell_E];
78 shell_an = [shell_an;shell_a];
79
80 if e < cold_limit && flag == 4 % För att spåra kalla elektroner
81 E0_count = E0_count+1;
82 E0_last(E0_count,:) = store_pos(end,:); % spara sista pos
83 E0_first(E0_count,:) = store_pos(1,:); % spara första pos
84 end
85 if flag == 1 % För att spåra escape-elektroner
86 Esc_count = Esc_count+1;
87 Esc_first(Esc_count,:) = store_pos(1,:); % spara första pos
88 end
89 first_A(i,:) = A; % För att spåra elektroner
90
91 if i == round(n/4)
92     disp('25% done')
93 elseif i == round(n/2)
94     disp('50% done')
95 elseif i == round(3*n/4)
96     disp('75% done')
97 end
98
99 mark = cumsum(marker);
100
101 p1 = F == 1;
102 p2 = F == 2;
103 p3 = F == 3;
104 p4 = F == 4;
105
106 C = categorical(F,[1 2 3 4],{'Escaped','Lost energy','Collided with ... 
    comet','Timed out'}); % Till histogram
% disp([num2str(i), ' simulations run'])
% disp([num2str(sum(p1)), ' escaped'])
% disp([num2str(sum(p2)), ' ran out of energy'])
% disp([num2str(sum(p3)), ' collided with comet'])
% disp([num2str(100*E0_count/n), '% cold electrons'])
% disp([num2str(i_count) ' total ionizations'])

figure('OuterPosition', [1030 550 580 530])
histogram(C, 'BarWidth', 0.5)
title(['Fate of particles, out of ', num2str(n), ' total'])
ylabel('Number of particles')

[val, idx] = max(full_E);
if any(idx == mark)
    numbr = idx == mark;
else
    numbr = idx < mark;
end
numb = find(numbr, 1) - 1;
figure('OuterPosition', [0 70 800 700])
% Användarskapad funktion, by Georg Stillfried
color_line3(Sx(mark(numb)+1:mark(numb+1))./1000, Sy(mark(numb)+1:mark(numb+1))./1000, Sz(mark(numb)+1:mark(numb+1))./1000, full_E(mark(numb)+1:mark(numb+1)));
c = colorbar;
c.Label.String = 'Energy [eV]';
hold on
plot3(0,0,0,'o','MarkerSize',10)
grid on
axis equal
title('Path of the most enegetic particle')
xlabel('x [km]')
ylabel('y [km]')
zlabel('z [km]')
az=45;
el = 45;
view(az,el) % Vrid kameran

if E0_count > 0
    figure('OuterPosition', [320 40 660 1040])
    subplot(2,1,1)
    plot3(E0_last(:,1)./1000, E0_last(:,2)./1000, E0_last(:,3)./1000, '*')
    hold on
    plot3(0,0,0,'o','MarkerSize',10)
    theta = linspace(0,2*pi);
x = stop*cos(theta);
y = stop*sin(theta);
plot(x./1000, y./1000) % Rita ut satellitens avstånd
    title(['Last position of cold electrons....', num2str(length(E0_first)),' particles'])
xlabel('x [km]')
ylabel('y [km]')
zlabel('z [km]')
% Vilket avstånd skapades de elektroner som blev kalla på?
E0_first_r = zeros(1,length(E0_first));
Esc_first_r = zeros(1,length(Esc_first));
first_A_r = zeros(1,length(first_A));
edges = 2:150;
for i = 1:length(E0_first_r)
    E0_first_r(i) = norm(E0_first(i,:));
end
for i = 1:length(Esc_first)
    Esc_first_r(i) = norm(Esc_first(i,:));
end
for i = 1:length(first_A)
    first_A_r(i) = norm(first_A(i,:));
end
h_0 = histcounts(E0_first_r./1000,edges);
h_esc = histcounts(Esc_first_r./1000,edges);
h_A = histcounts(first_A_r./1000,edges);
qu_esc = h_esc./h_A*100;
qu_cold = h_0./h_A*100;
figure
plot(qu_cold)
title('Probability that electron becomes cold as function of how far from nucleus it is created')
xlabel('Distance from nucleus [km]')
ylabel('Probability [%]')
grid on

figure
plot(qu_esc)
title('Probability that electron escapes as function of how far from nucleus it is created')
xlabel('Distance from nucleus [km]')
ylabel('Probability [%]')
grid on
dens_ind = zeros(1,shells);
E_T = zeros(1,shells);
nr_counter = zeros(1,shells);

str_1 = 'Energy distribution at %d km,';
str_2 = '%d hits';

if shells < 60
  % Avgör vilka dimensioner subploten ska ha
  f = factor(shells+1);
  f_size = size(f);
  if f_size(2) == 1 % Primtal
    f = factor(shells+2);
    f_size = size(f);
  end
  if mod(f_size(2),2) == 0 % Jämnt antal
    size_a = prod(f(1:f_size(2)/2));
    size_b = prod(f((f_size(2)/2)+1:end));
  else % Udda antal
    size_a = prod(f(1:floor(f_size(2)/2)));
    size_b = prod(f(ceil(f_size(2)/2):end));
  end
  figure('OuterPosition',[800 70 1100 750])
else
  % Mindre subplot, 5*2
  size_a = 2;
  size_b = 5;
  figure('OuterPosition',[200 70 1600 750])
end

countt = 0;
countz = 0;
for_surf_val = zeros(10,20);
for_surf_edge = zeros(10,20);
for_surf_dist = zeros(10,20);
miniN = zeros(1,10);
miniplots = [];

for q = 1:shells
  l = shell_En(:,q);
  L = l(~isnan(l));
  alpha = shell_an(:,q);
  Alpha = alpha(~isnan(alpha));
  new_factor = min(1./abs(cos(Alpha)),max_factor);
  dens_ind(q) = sum(new_factor./(sqrt(L)*targets(q).^2)); % ... täthetsindikatorn
  E_T(q) = mean(L);
  nr_counter(q) = length(L);
  if n > 20
    N = 20;
  elseif n < 3
    N = 2;
  elseif length(L) > 20
    N = 20;
  else
    N = ceil(length(L)/1.5);
  end
  if shells < 60

```matlab
subplot(size_a,size_b,q)
histogram(L,N)
s1 = round(targets(q)/1000);
s2 = length(L);
str_a = sprintf(str_1,s1);
str_b = sprintf(str_2,s2);
title([str_a,str_b])
xlabel('Energy [eV]')
ylabel('Number of particles')
else
    if any(q == floor((1:10)*shells/10))
        countt = countt + 1;
        subplot(size_a,size_b,countt)
        h = histogram(L,N);
s1 = round(targets(q)/1000);
s2 = length(L);
str_a = sprintf(str_1,s1);
str_b = sprintf(str_2,s2);
title([str_a,str_b])
xlabel('Energy [eV]')
ylabel('Number of particles')
for_surf_dist(countt,:) = s1;
for_surf_edge(countt,:) = h.BinEdges(2:end);
for_surf_val(countt,:) = h.Values;
elseif any(q == 1:10)
    countz = countz + 1;
    if countz > 2
        if length(L) > length(miniplots(countz-1,:))
            miniplots(countz-1,length(L)) = 0;
        end
    elseif countz == 2
        miniplots(1,length(L)) = 0;
    end
    miniplots(countz,:) = L';
    miniN(countz) = N;
end
end
end
end
figure('OuterPosition',[200 70 1600 750])
for i = 1:10
    subplot(2,5,i)
    h = histogram(miniplots(i,:),miniN(i));
s1 = round(targets(i)/1000);
s2 = length(miniplots(i,:));
str_a = sprintf(str_1,s1);
str_b = sprintf(str_2,s2);
title([str_a,str_b])
xlabel('Energy [eV]')
ylabel('Number of particles')
end
end
end
end
figure
surf(for_surf_dist,for_surf_edge,for_surf_val)
set(gca, 'ydir','reverse')
axis square
grid on
title('Energy distribution over distance')
xlabel('Distance [km]')
ylabel('Energy [eV]')
```

347 zlabel('Number of particles')
348 if shells < 60
349 subplot(size_a,size_b,shells+1)
350 else
351 figure('OuterPosition',[1200 70 600 550])
352 end
353 if n > 20
354 N = 20;
355 elseif n < 3
356 N = 2;
357 elseif length(fin_E) > 20
358 N = 20;
359 else
360 N = ceil(length(fin_E)/1.5);
361 end
362 histogram(fin_E, N)
363 s2 = length(fin_E);
364 str_3 = '%d particles';
365 str_c = sprintf(str_3,s2);
366 title({'Final energies of particles,',str_c})
367 xlabel('Energy [eV]');
368 ylabel('Number of particles')
369
370 figure('OuterPosition',[1200 530 600 550])
371 l = shell_En(:,sat_shell);
372 L = l(~isnan(l));
373 histogram(L,N)
374 s3 = length(L);
375 str_4 = '%d particles';
376 str_d = sprintf(str_4,s3);
377 s4 = round(targets(sat_shell)/1000,1);
378 str_5 = '%g km ';
379 str_e = sprintf(str_5,s4);
380 title({'Energies of particles at' str_e, str_d})
381 xlabel('Energy [eV]');
382 ylabel('Number of particles')
383
384 figure
385 plot(targets./1000,E_T)
386 title('Mean energy of particles')
387 xlabel('Distance [km]')
388 ylabel('Mean energy [eV]')
389
390 n_e = dens_ind;
391 T = E_T;
392 p = n_e.*T;
393 grad_p = gradient(p,targets);
394 E_rad = -grad_p./(n_e);
395
396 r_c = 2e3; % Radien av den kemeten, [m]
397 if isequal(E_kind,'Fixed')
398 E_amb = 10.*{(targets-2*r_c)./(targets.*(targets-r_c))}; % ...
399 Fältstyrka ambipolärt, minusstecken?
400 elseif isequal(E_kind,'None') || isequal(E_kind,'Only from comet')
401 E_amb = zeros(1,shells);
else % if E_kind == 'Custom'
    E_amb = zeros(1,shells);
    for o = 1:shells
        E_amb(o) = E_find(targets(o));
    end
end
E_r = E_amb;
figure('OuterPosition',[650 40 550 500])
plot(targets./1000,E_rad,targets./1000,E_r)
title('E-field')
xlabel('Distance [km]')
ylabel('Field strength [V/m]')
grid on
legend('Calculated','Used')
if any(isnan(E_rad))
    disp('Some values of E_rad are NaN, ignoring those values')
    indx = sum(isnan(E_rad))+1;
    E_rad = E_rad(indx:end);
end
fourier = fft(E_rad);
P2 = abs(fourier/shells);  
P1 = P2(1:shells/2+1);  
P1(2:end-1) = 2*P1(2:end-1);  
Fs = shells;
f = Fs*(0:(shells/2))/shells;
figure
plot(f,P1)
title(['Single-Sided Amplitude Spectrum of E_{rad}, ',num2str(n),' ... particles'])
xlabel('f')
ylabel('|P1(f)|')
if quest == 'Y'
    if any(isnan(E_rad))
        disp('Warning! Calculated E-field not usable because some ... values are NAN.')
        disp('Please re-run the simulation.')
    else
        save('E_rad_sav','E_rad','targets')
    end
end
Core loop, newton_sim

% Denna funktion simulerar banan för en partikel
function [flag, count, store_pos, store_E, E, shell_E, shell_a, ... 
    store_s] = newton_sim_clean(A,Z,E,stop,targets,E_kind,time_limit)
% Startvillkor
zer = -A./norm(A); % Enhetsvektor för positionen, är oskulle r_hat
t_num = 0;
r = sqrt(A(1).^2+A(2).^2+A(3).^2);  
T = 10;

r_c = 2e3; % Radien av kometens, [m]
Q = 2*10^28; % Antalet partiklar som lämnar kometytan (per sekund?)
v = 1000; % Partiklarnas hastighet i m/s
N_a = Q/(4*pi*v); % = Neutralställen + r^2
q = 1.6021766*10^-19; % Elektronladdning Coulomb

count = 1;
flag = 0; % Om flag /= 0 stannar simuleringen
trip_count = zeros(1,length(targets));
old_trip = r > targets; % Vilka är större

store_pos = [];
store_E = [];
shell_E = nan(1,length(targets));
store_s = [];
shell_a = nan(1,length(targets));

while flag == 0
    store_pos(count,:) = A;
    store_E(count) = E;
    % ta steg, tidsteg ca 10 microsekunder
    sig = sigma_Iitikawa(E)*10^-4; % konvertera till m^2
    if r > 2e3+200
        A_temp = A + 10*Z;
        r_min = norm(cross(A_temp - A,A))/norm(A_temp - A); % Närmaste punkten
        if r_min > r % Så vi aldrig kan komma under ytan
            r_min = r;
        elseif r_min < r_c
            r_min = r_c;
        end
        s = 1000-3.9600e+09/(r_min)^2;
    else
        s = 1;
    end
    store_s(count) = s;
    v = E_to_v(E); % m/s
    t = s/v;
    t_sum = t_sum + t;
    P = 1-exp(-sig.*n_N./(r.^2).*s);
    u = rand;
    if P > u % Interaction sker
        [Z,E] = inter_function(Z,E,sig);
    else % Interaction sker ej
        V = 2*v;
    end
    if isequal(E_kind,'Fixed')
        E_amb = T*(r-2*r_c)/(r*(r-r_c)); % Fältstyrka ambipolärt
    elseif isequal(E_kind,'None')
        E_amb = 0;
    else % if E_kind == 'Custom'
        E_amb = E_find(r);
    end
    E_r = E_amb;
    F = E_r*q; % Newton
    m = 9.109383*10^-31; % kg
    a = F/m; % m/s^2, acceleration alltid i anti-radiell riktning
    dv = a*t; % förändring av hastighet
    Dv = zer*dv;
    V = V + Dv;
    Z = V./norm(V);
    E = v_to_E(norm(V));

76 \text{S = V*t; } % Steget som tas
79 \text{end}
82 \text{zer} = -A./\text{norm}(A);
85 \text{r} = \text{norm}(A);
88 \% Läs av energi i olika skal
90 \text{trip} = r > \text{targets};
92 \text{if isequal(trip, old_trip)}
95 \% Pass
98 b = \text{max(sum(trip),sum(old_trip))};
101 \text{trip\_count(b)} = \text{trip\_count(b)}+1;
104 \text{M, } \text{~} = \text{size(shell\_E)};
107 \text{if trip\_count(b) > M}
110 \text{shell\_E} = [\text{shell\_E}; \text{nan}(1,\text{length(targets)})];
113 \text{shell\_a} = [\text{shell\_a}; \text{nan}(1,\text{length(targets)})];
116 \text{shell\_E}(\text{trip\_count(b)},b) = E;
119 \text{shell\_a}(\text{trip\_count(b)},b) = \cos(\text{dot}(a,b)/(\text{norm}(a)*\text{norm}(b)));
122 \text{old\_trip} = \text{trip};
125 \% Kolla om vi ska fortsätta
128 \% if start\_r < sat \% elektronen startar innanför satelliten
131 \% if out == false \% elektronen startar innanför satelliten
134 \text{if r > stop}
137 \text{flag} = 1; \% Electron escaped
140 \text{else if E > 60 \% Om energin är hög nog, avsluta banan. Ska ... egentligen vara 3*(genomsnittsenergin).}
143 \text{flag} = 3; \% Electron collided with comet
146 \text{else if r < r\_c-2}
149 \text{flag} = 3; \% elektronen fastnar?
152 \text{else \% Om energin är låg, elektronen studsar}
155 Z = Z - 2*(Z*\text{zer}')*\text{zer};
158 \text{end}
161 \text{end}
164 \text{if E < 0.001}
167 \text{r\_t} = \text{abs}(r-2*\text{r\_c});
170 \text{if r\_t < 100}
173 \text{flag} = 2; \% Electron lost its energy
176 \text{else}
179 E = 0.002;
182 \text{end}
185 \text{end}
188 \text{count} = \text{count} + 1;
191 \text{if count > 7e5 || t\_sum > time\_limit}
194 \text{flag} = 4; \% Simulation timed out
197 \text{end}
199 \text{end}
202 \text{store\_s} = \text{cumsum(store\_s)};
205 \text{end}

\text{Interaction function, } \text{inter\_function}
function [X,e] = inter_function_clean(Z,E,sig)
% Sig är summan av alla cross sections (elast + inelast + elektr + ... ion).
int = 4; % antal olika interaktioner som finns
global i_count
Q = zeros(1,int);
if E < 20
    sig_v = sigma_vib_tot(E)*10^-4; % Konvertera till m^2, normalt
else % E > 20
    s = [0.3200 0.1800]; % linjär approx utifrån sista två punkterna
    s = s * 10^-20; % cm^2 -> m^2
    e = [15.0000 20.0000];
    m = s(1)-a*e(1);
    % p = [a m];
    sig_v = m + E * a;
    sig_v = sig_v*1.2;
    if sig_v < 0
        sig_v = 0;
    end
    sig_i = sigma_ion_Itikawa(E)*10^-4;
    sig_e = sigma_electr_tot_Itikawa(E)*10^-4;
    sig_d = sigma_diss_Itikawa(E)*10^-4;
    Q(1) = (sig-sig_v)/sig; % elast + ion / tot, cross section för inelast
    Q(2) = (sig-sig_i)/sig;
    Q(3) = (sig-sig_e)/sig;
    Q(4) = (sig-sig_d)/sig;
    P = (1-Q)./int;
    P = cumsum(P);
    P = [P 1];
    u = rand;
    p = P > u;
    reac = find(p,1);
    if reac == 1
        % inelastisk
        c = pi/36; % Vinkeln för rotation kring z, 5 grader, fast vinkel
        if E < 20 % Ful lösnings, förhindrar fel vid stora energier
            q1 = sigma_vib(E,0);
            q2 = sigma_vib(E,1);
        else
            q1 = 0.05 * 10^-16;
            q2 = 0.04 * 10^-16;
        end
    q = q1 / (q1+q2);
    vibr = rand;
    if vibr < q % Energiförlust
        e = E-0.453; % vibs
    else
        e = E-0.198; % vibs
    end
    elseif reac == 2 % Jonisering
    i_count = i_count+1;
    c = angle_Itikawa;
\[ e = E - \text{energy\_loss\_ion\_alt}(E); \] % Energiförlust

```plaintext
e = E - \text{min}(10, 0.75 \times E));
```

```plaintext
e = E - (5.2 + \text{rand} \times 2.5); % Kommer inte att ske om energin är > 7.8
```

```plaintext
e = E - 0.004; % Mycket liten energiförlust, 4 meV
```

```plaintext
e = 0;
```

```plaintext
V = [1,0,0]; % bas
```

```plaintext
M = \text{vrotvec2mat}(\text{vrotvec}(Z,V)); % Hitta vridningen
```

```plaintext
R_z = [\cos(c), \sin(c), 0; -\sin(c), \cos(c), 0; 0, 0, 1];
```

```plaintext
V = V R_z;
```

```plaintext
V = V R_x; % slumpvis rotation kring x
```

```plaintext
X = V \times M;
```

```plaintext
function [sigma] = \text{sigma\_Itikawa}(E)
% Tabell 3
```

```plaintext
e = [1 1.2 1.4 1.6 1.8 2.2 2.5 2.8 3.1 3.4 3.7 4 4.5 5 5.5 6 6.5 ... 7 7.5 8 ... 8.5 9 9.5 10 11 12 13 14 15 16 17 18 19 20 22 25 30 35 40 50 60 ... 70 80 90 100 ... 120 150 200 250 289 361];
s = [110 95.3 82 71 62.3 54.2 51.1 46.9 43.2 39.8 37.2 34.8 33.5 ... 31.4 30.2 ... 29.1 28.4 27.3 26.8 26.5 25.8 25.5 24.8 23.7 23.2 22.8 22.4 ... 21.7 21.0 20.3 ... 19.6 19.1 18.6 18.3 17.7 16.9 15.6 14.1 13.1 12.2 10.5 9.7 8.9 ... 8.3 7.7 7.1 ... 6.5 5.6 4.8 4.2 3.78 3.19];
s = s \times 10^{-16}; % \text{cm}^2
```

```plaintext
if any(E == e) % om energin stämmer exakt
    sigma = s(find(e == E, 1));
else % Linear approx
    l = find(e > E, 1);
    if l == 1
        x = e(l-1); 
        y = s(l-1);
    elseif isempty(l) % Not good, fail safe
```

Total cross-section look-up, sigma\_Itikawa
function [sigma] = sigma_electr_tot_Itikawa(E)
% konstanterna e och s beräknade i "const_sigm_electr.mat"

e = [20.0000 25.0000 30.0000 37.5000 40.0000 60.0000 ... 62.5000 ... 70.0000 75.0000 80.0000 87.5000 90.0000 100.0000];
s = [0.2040 1.2280 2.3186 3.8000 4.3160 6.2900 8.4400 8.8975 ... 9.9820 10.6300 11.1140 11.7500 11.8140 12.0000];
s = s * 10^-18; % cm^2

if any(E==e) % om energin stämmer exakt
sigma = s(find(e==E,1));
else % Linear approx
l = find(e>E,1);
if l == 1
x = e(l:l+1);
y = s(l:l+1);
elseif isempty(l)
x = e(end-1:end);
y = s(end-1:end);
else
x = e(l-1:l);
y = s(l-1:l);
end

a = (y(2)-y(1))/(x(2)-x(1));
m = y(1)-a*x(1);
sigma = m + E * a;
end
if sigma < 0
sigma = 0;
end

function [sigma] = sigma_diss_Itikawa(E)
% Tabell 24

e = [10 15 20 30 50 75 100 150];
s = [0.15 0.48 0.7 1.3 1.9 2.1 2.05 1.98];
s = s * 10^-18; % cm^2

if any(E==e) % om energin stämmer exakt
sigma = s(find(e==E,1));
else % Linear approx
l = find(e>E,1);
x = e(end-1:end);
y = s(end-1:end);
a = (y(2)-y(1))/(x(2)-x(1));
m = y(1)-a*x(1);
sigma = m + E * a;
end
if sigma < 0
sigma = 0;
end

Cross-section look-up, electric interaction, sigma_electr_tot_Itikawa

Cross-section look-up, dissociation, sigma_diss_Itikawa
if \( l == 1 \)
\[
x = e(l:l+1);
\]
\[
y = s(l:l+1);
\]
elseif isempty(l)

% Extrapolera utifrån sista elementen, kanske inte fysiskt ...

korrekt
\[
x = e(end-l:end);
\]
\[
y = s(end-l:end);
\]
else
\[
x = e(l-1:l);
\]
\[
y = s(l-1:l);
\]
end

a = \((y(2)-y(1))/(x(2)-x(1))\);
\[
m = y(1)-a*x(1);
\]
\[
sigma = m + E * a;
\]
end
if sigma < 0
\[
sigma = 0;
\]
end
end

Cross-section look-up, vibration, sigma_vib_tot

function [sigma] = sigma_vib_tot(E)
if E < 0.198
sigma = 0;
else
% konstanterna e och s var beräknade i "const_sig_vib.mat"
\[
e = [0.1980 0.3000 0.3900 0.4530 0.5300 0.5800 0.6000 ... 0.6300 ... 0.8000 0.8620 1.0000 2.0000 2.1000 2.2000 3.0000 4.0000 ... 5.0000 ... 6.0000 7.0000 7.5000 8.0000 8.8750 10.0000 15.0000 20.0000];
\]
\[
s = [0 1.7100 1.8050 1.7840 1.4853 3.3703 3.2033 3.0885 3.0139 ... 1.0912 ... 0.9680 0.8700 0.5364 0.5150 0.5089 0.4700 0.5350 ... 0.5870 0.6520 0.6865 0.6972 0.6650 0.5874 0.5050 0.3200 ... 0.1800];
\]
\[
s = s * 10^{-16}; % cm^2
\]
if any(E==e) % om energin stämmer exakt
sigma = s(find(e==E,1));
else % Linear approx
\[
l = find(e>E,1);
\]
\[
x = e(l-1:l);
\]
\[
y = s(l-1:l);
\]
\[
a = (y(2)-y(1))/(x(2)-x(1));
\]
\[
m = y(1)-a*x(1);
\]
\[
sigma = m + E * a;
\]
end
end

Cross-section look-up, modes of vibration, sigma_vib

function [sigma] = sigma_vib(E,mode)
if mode == 1
% 010
e = [0.198 0.3 0.34 0.39 0.6 0.862 1 2 2.2 3 4 5 6 8 10 15 20];
s = [0 1.71 1.805 1.784 0.7885 0.399 0.37 0.2 0.19 0.16 0.15 ... 
0.157 0.163 0.17 0.18 0.13 0.1];

else % mode = 2
  e = [0.453 0.53 0.58 0.63 0.8 1 2.1 3 4 5 6 7 7.5 8 8.875 10 15 ... 
20];
s = [0 2.25 2.32 2.27 0.6 0.5 0.32 0.31 0.385 0.43 0.489 0.52 ... 
0.529 0.495 0.413 0.325 0.19 0.08];
end

s = s * 10^-16; % cm^2

if any(E==e) % om energin stämmer exakt
  sigma = s(find(e==E,1));
else % Linear approx
  l = find(e>E,1);
  if l == 1
    sigma = 0; % Mindre än minimi-värde
  elseif isempty(l)
    % Extrapolera utifrån sista elementen, kanske inte fysiskt ...
    korrekt.
    x = e(end-1:end);
    y = s(end-1:end);
  else
    x = e(l-1:l);
    y = s(l-1:l);
  end
  a = (y(2)-y(1))/(x(2)-x(1));
  m = y(1)-a*x(1);
  sigma = m + E * a;
end

if sigma < 0 % failsafe
  sigma = 0;
end

Cross-section look-up, ionization, sigma_Ion.Italicawa

function [sigma] = sigma_Ion_Itikawa(E)
% Tabell 3
e = [13.5 15 17.5 20 22.5 25 30 35 40 45 50 60 70 80 90 100 110 125 ... 
    150 175 200 250];
s = [0.025 0.126 0.274 0.428 0.609 0.761 1.02 1.26 1.43 1.59 1.72 ... 
    1.88 1.99 ... 
    2.09 2.13 2.16 2.15 2.13 2.05 1.99 1.9 1.73];
s = s * 10^-16; % cm^2
if any(E==e) % om energin stämmer exakt
  sigma = s(find(e==E,1));
else % Linear approx
  l = find(e>E,1);
  if l == 1
    x = e(l:l+1);
    y = s(l:l+1);
  elseif isempty(l)
    % Extrapolera utifrån sista elementen, kanske inte fysiskt ...
    korrekt.
    x = e(end-1:end);
    y = s(end-1:end);
  else
    x = e(l-1:l);
    y = s(l-1:l);
  end
  a = (y(2)-y(1))/(x(2)-x(1));
  m = y(1)-a*x(1);
  sigma = m + E * a;
end
if sigma < 0 % failsafe
  sigma = 0;
end
end
Energy loss from ionization, energy_loss_ion_alt

1 function [e] = energy_loss_ion_alt(E)
2
3 e_s = rand(1,10).*(E-12.6);
4 F = (1+e_s/13).^(-2.1);
5 R = cumsum(F);
6 norm = 1/R(end);
7 R = R*norm;
8 u = rand;
9 r = R > u;
10 l = find(r,1);
11 12 E_s = e_s(l);
13 e = E_s+12.6;
14
15 end

Scattering angle, angle_Itikawa

1 function [S] = angle_Itikawa() % Gäller för en elastisk kollision ...
2 % vid 6 eV.
3 ang = 0:10:180; % vinkel i grader
4 ang(1) = 1; % Fixa till det första elementet
5 ang = ang.*pi./180; % gör om grad till rad
6 s = [100, 11, 4.8, 2.5, 1.5, 1.2, 1.1, 1, 0.82, 0.69, 0.48, 0.4, ...
7 0.35, 0.38, 0.45, 0.55, 0.67, 0.8, 0.82]; % diff-cross section ...
8 i 10^-16 cm^2/sr
9 c = cumsum(s);
10 normal = 1/c(end); % "Normaliering", så att sannolikheten går mot 1
11 c = c.*normal;
12 u = rand;
13 p = c > u;
14 l = find(p,1);
15 S = ang(l);
16 end