Report # 6+7

# Summary of BOLOS.

In general, both Bolsig+ and Bolos based on the method in paper of *Hagelaar et Pitchford* [1]. I read this paper, and compared with source-code of Bolos library. I point out some difference of operating condition between Bolsig+ and Bolos, as well as show which term was used in work of *HongTao Zheng* [2] and *Robby Aerts* [3] in Table 1.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Bolsig+ | Bolos | Hongtao Zheng and Robby Aerts |
| Gas-phase Composition | Yes | Yes | Yes |
| Electron – Neutrals and Radicals collision | Yes | Yes | Yes |
| Electron – electron collision (need Ionization degree) | Yes | No | Is not stated |
| Electron – Ion collision (need Ionization degree) | Yes | No | Is not stated |
| Super-elastic (need Excited state temperature) | Yes | No | Is not stated |
| AC field (need Frequency) | Yes | Only DC | AC field |

Table 1 Term is taken into account in Bolsig+ and Bolos, which one was used in work of Hongtao Zheng and Robby Aerts

After simplifying the Boltzmann equation with some assumption, Bolsig+ solve the equation of electron energy distribution function , which take into account all collision term (*Equation 39* in reference [1])

|  |  |  |
| --- | --- | --- |
|  |  |  |

Where

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

The terms contain are terms regarding to electron-electron collision (*section 2.4* in reference [1]).

Bolos solve the same equation of electron energy distribution function , but it doesn’t take account the e-e collision. I checked it in *solve.py* of Bolos.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

\*NOTE: To use Bolos for our simulation, we need to add the feature AC field into Bolos.

# Add AC field to Bolos

From the *equation 25* of ref. [1], I tried to add the feature of AC field into account of Bolos. I created a new *solve2.py* base on solve.py of Bolos, and the results of EEDF (electron energy distribution function) is reasonably compared with Bolsig+.

I tested this code with operating parameters in table 2, and results in table 3

|  |  |
| --- | --- |
| Parameters | Value |
| Gas Temperature (K) | 400 K |
| Gas Composition | CH4:0,5 – Ar:0.5 |
| Electric field / N (Td) | 100 Td 900 Td |
| Angular Frequency (m3.rad/s) | 0.10000E-11 |

Table 2 Operating Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Electric field / N (Td) | Mean Electron Energy (eV) | | Error |
| Bolsig+ | Our test |
| 100 | 0.0948 | 0.0945 | 0.32% |
| 200 | 0.114 | 0.1146 | 0.53% |
| 300 | 0.170 | 0.1457 | 14.29% |
| 400 | 0.472 | 0.4926 | 4.36% |
| 500 | 1.21 | 1.0041 | 17.02% |
| 600 | 2.10 | 2.0705 | 1.40% |
| 700 | 2.88 | 2.8720 | 0.28% |
| 800 | 3.49 | 3.4931 | 0.09% |
| 900 | 4.01 | 4.0079 | 0.05% |
| 1000 | 4.46 | 4.4587 | 0.03% |
| 1100 | 4.87 | 4.8711 | 0.02% |
| 1200 | 5.26 | 5.2586 | 0.03% |
| 1300 | 5.63 | 5.6300 | 0.00% |
| 1400 | 5.99 | 5.9904 | 0.01% |
| 1500 | 6.34 | 6.3435 | 0.06% |
| 1600 | 6.69 | 6.6914 | 0.02% |
| 1700 | 7.03 | 7.0357 | 0.08% |
| 1800 | 7.37 | 7.3778 | 0.11% |
| 1900 | 7.71 | 7.7186 | 0.11% |
| 2000 | 8.05 | 8.0587 | 0.11% |

Table 3 The comparison between Bolsig+ and our code

The plots of EEDF are below:

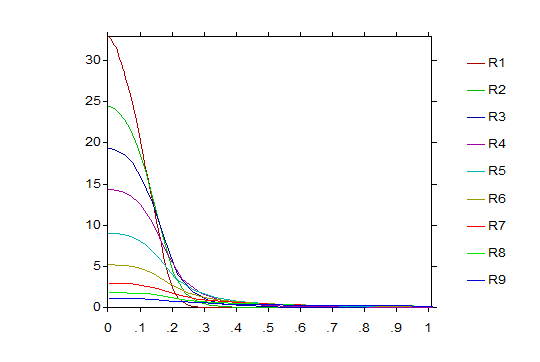


Figure 1 EEDF of Bolsig+ from 100Td to 900 Td

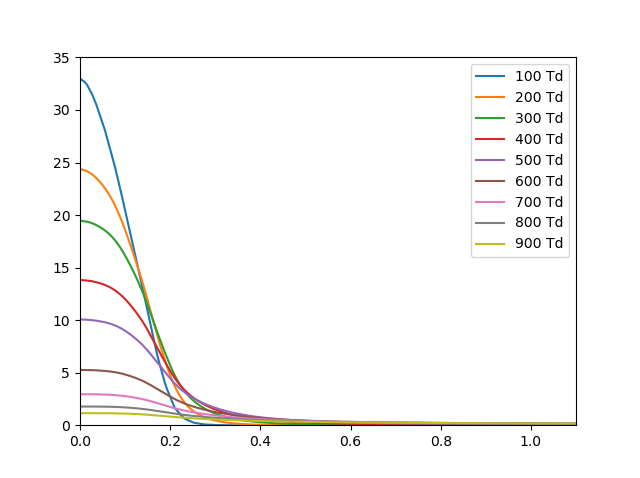


Figure 2 EEDF of our code from 100Td to 900 Td.

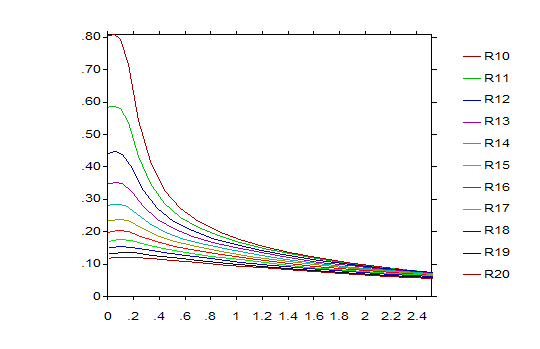


Figure 3 EEDF of Bolsig+ from 1000Td to 2000 Td

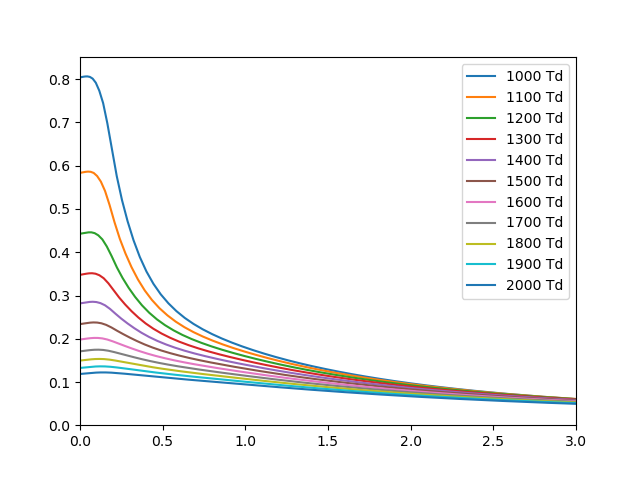
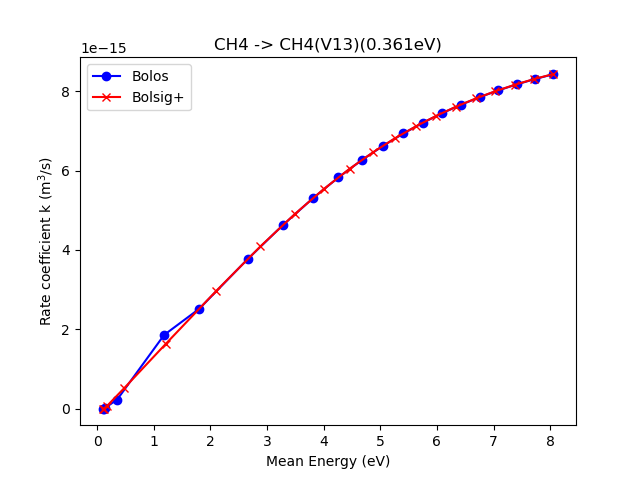
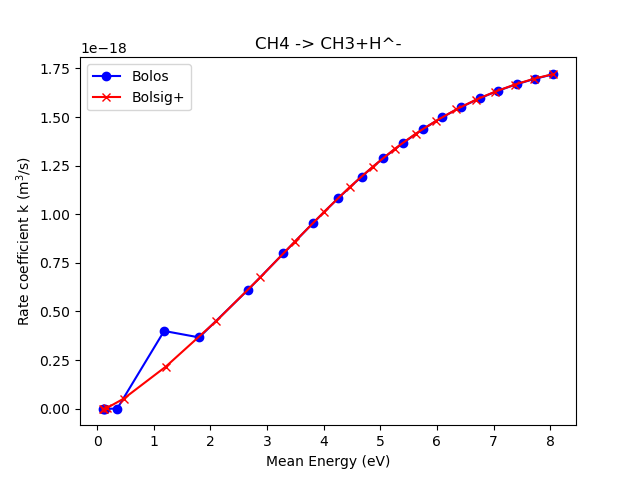


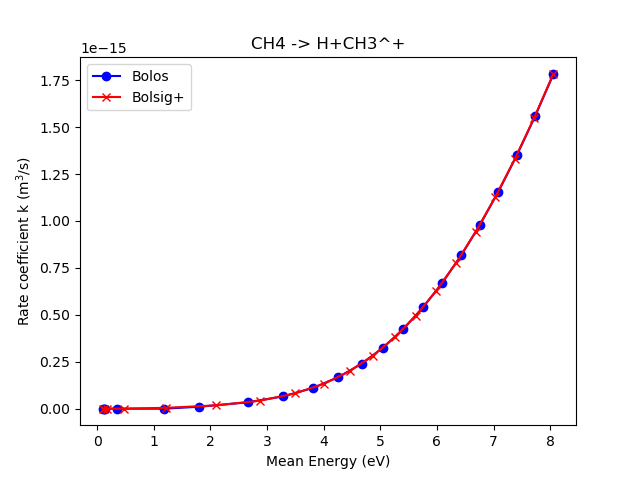
Figure 4 EEDF of our code from 1000Td to 2000 Td

\*Conclude: In general, our code could take the AC field into account, and there is big error of mean electron energy in range of Electric field / N from 300 Td – 500 Td.

However, we can use it to calculate the rate coefficients, the result only has error in the range of low mean energy in some case. I point out some case in the figure below.







# Fit curve code

After we have the data of rate coefficients, we need to convert it into a common form and easy to use in Cantera. Therefore, I wrote a script to get Arrhenius form of rate coefficients, using *scipy.optimizer.fit\_curve()* which based on the method of least squares of error.

First time, I used directly Arrhenius equation (Eq. 6), however, in some case, iterative solver of *scipy.optimizer.fit\_curve()* reach maximum iteration and doesn’t get the convergence value. Because the initial value (guess value) is not good.

|  |  |  |
| --- | --- | --- |
|  | Or |  |

Second time, I convert the Arrhenius equation into logarithm form (Eq. 7), the solver converge rapidly, but the result is not good (Fig. 5).

|  |  |  |
| --- | --- | --- |
|  |  |  |

Finally, I use hybrid-method. Firstly, I used the log logarithm form to get rapidly the reasonable guess value, then I used original Arrhenius equation to fit. The final result is very good !

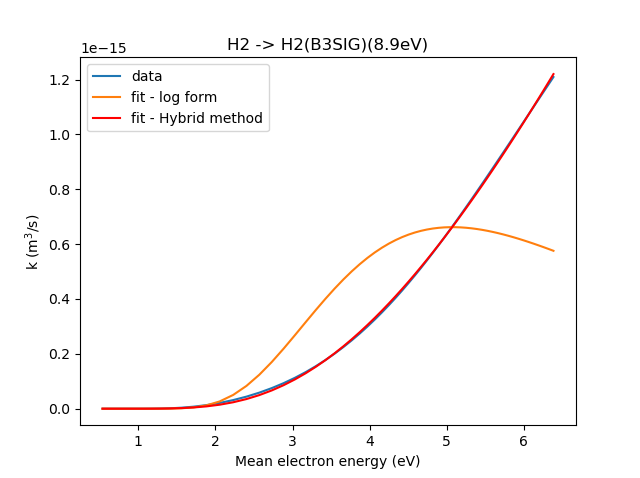


Figure 5 Fitting curve

# Governing Equation in 0-D plasma model

We consider a 0D reactor in a volume , without surface-reaction.

* Global Mass balance Equation:

Integrate in all volume V, we have

Where: is mass flow at inlet, is mass flow at outlet.

* Species Conservation Equation:

Integrate in all volume V, we have

Where: is molar reaction rate of species k, is atomic weight of species k, is mass fraction at inlet of species k, is mass fraction at outlet of species k.

* State Equation
* Electron Energy Equation:

The transient form of the electron energy balance equates the time-rate-of-change of the electron swarm’s internal energy , to the net flow of electron enthalpy into and out of the reactor, accounting for net chemical production rates, collisions losses , and power deposition from externally applied electromagnetic fields [4].

Where:

* Electron density: and Mass Flow rate of electron
* , Refers to the electron enthalpy of newly created electrons in the gas-phase, when electrons are formed from the ionization of a relatively cold neutral, the electron is assumed to originate close to the neutral temperature.
* Molar reaction rate of creating electron from ionization reactions.
* Internal Energy
* Heat capacity

Ep.9 is rewritten:

Take (Eq.10) subtract (Eq.8) multiplied by for electron, we have a equation same as Equation (4) of Hongtao’s paper. However, *Hongtao et al.* didn’t take the energy required to thermalize new electrons into account.

Where:

. With is the momentum-transfer collision frequency between the electrons and the kth heavy species. The plasma-reactor model calculates the momentum-transfer collision frequencies from momentum-transfer collision cross sections specified [4]. I think Bolsig+ and Bolos could calculate it

, represents the summation of electron energy loss per electron-impact reactions. The total number of electron-impact (i.e., electron-temperature dependent) reactions , is the net rate of progress of the r-th reaction (this is rate coefficient calculated from Bolsig+ or Bolos for inelastic reactions), is the net enthalpy change of the reaction, which can be determined from species’ thermochemistry.

# Conclusion

Up till now, we have had a program to solve Boltzmann equation, which can take AC electric field into account. However, It doesn’t take effects of e-e collisions (an option in Bolsig+). Moreover, I am not sure that the authors Hongtao Zheng and Robby took or not the e-e collisions into account in their work.

We could calculate the rate coefficients of electron – impact reactions and these result is reasonable compared with Bolsig+. We also could convert these rate coefficient into Arrhenius form.

\*NEXT WORKS:

- I will continue find the way to determine the unknown parameters of Energy Equation e.g. , .

- I will focus on writing a program to solve Electron Energy Equation.

\*I also would need your advice for my next works.

# References

|  |  |
| --- | --- |
| [1] | G. J. M. Hagelaar, L. C. Pitchford, "Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models," *Plasma Sources Science and Technology,* no. 14, p. 722, 2005. |
| [2] | Liu, Hongtao Zheng and Quian, "Kinetic Study of Nonequilibrium Plasma-ssisted Methane Steam Reforming," *Mathematical Problems in Engineering,* 2014. |
| [3] | Robby Aerts, Tom Martens, and Annemie Bogaerts, "Influence of Vibrational States on CO2 Splitting by Dielectric Barrier Discharges," *Physical Chemistry,* no. 116, pp. 23257-23273, 2012. |
| [4] | ANSYS Chemkin Theory Manual 17.0 (15151), Reaction Design, San Diego, 2015, p. 168. |

Link:

Gantt chart: <https://docs.google.com/spreadsheets/d/1pgTm9QTVgA2n_2NetLwpwnbW8kADJNP6VqgRpcv_keI/edit?usp=sharing>