

A GLOBAL MODEL FRAMEWORK WITH SELF-CONSISTENT ELECTRON
ENERGY DISTRIBUTION FOR REACTION KINETICS IN LOW-TEMPERATURE
PLASMAS

By

Janez Krek

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computational Mathematics, Science and Engineering – Doctor of Philosophy

2021

ABSTRACT

A GLOBAL MODEL FRAMEWORK WITH SELF-CONSISTENT ELECTRON ENERGY DISTRIBUTION FOR REACTION KINETICS IN LOW-TEMPERATURE PLASMAS

By

Janez Krek

Fundamental interactions in plasmas are very well defined and researched, but the sheer number of possible interactions between particles and materials makes plasma research very complicated. Processes in plasma occur over many orders of magnitude in time and spatial scale, requiring careful selection of physical and mathematical models for describing included processes. Geometry simplification and reduction of spatial complexity are made to include more detailed and complicated reaction kinetics (reaction sets). Global models (volume-averaged or 0D model), well known for their fast evaluation times, are often employed where spatial dependence of plasma parameters is not significant and more complicated chemistry is required.

The electron energy distribution function (EEDF) profoundly affects reaction kinetics as many reactions depend on the electron energy. The Maxwellian EEDF is often assumed, leading to overestimated electron impact reaction rates and time evolution of species densities. A self-consistent EEDF evaluation is incorporated in the global model framework to capture the temporal evolution of the EEDF and further increase the fidelity of the results. The EEDF evaluation frequency is user-definable and affects both simulation accuracy and run times. The Kinetic Global Model framework (KGMf), coupled with self-consistently evaluated EEDFs, has been used to study complex reaction chains in pulsed low-temperature plasmas across a range of pressures with good comparison to theoretical, computational, and experimental results from the literature.

Many input parameters in plasma simulations, i.e., impact cross-sections and the reaction rate coefficients, are derived from experiments and approximate theories, and they contain

uncertainties. Additionally, experimentally measured values from different experiments can deviate significantly. Sensitivity analysis can quantify the effect of input parameter uncertainties on plasma behavior for a given system. The reaction rate coefficients were selected as input parameters for global sensitivity analysis (GSA), and a high-dimensional input parameter space was reduced with Saltelli's sampling. The effects of reaction rate coefficient uncertainties on species densities in a nanosecond pulse discharge in an $\text{H}_2\text{-O}_2\text{-Ar}$ gas mixture were explored and quantified using sensitivity indices.

Large reaction sets are prohibitively expensive in spatially dependent simulations, limiting detailed chemistry in fluid and PIC simulations. Reactions and species have different contributions to the selected plasma quantities of interest, e.g., species density or temperature. Defining the importance of reactions and species for a given system can reduce the reaction set with known effects on the selected quantity of interest. The reduced reaction set has lower computational complexity and is suitable for global sensitivity analysis or spatially dependent simulations. The directed relation graph (DRG) method is presented and deployed in a nanosecond pulse discharge in an $\text{H}_2\text{-O}_2\text{-Ar}$ gas mixture, reducing the reaction set while preserving the H and H^+ densities. The reduction steps were compared in terms of the size of the reduced reaction set and the simulation run time reduction.

To my parents.

ACKNOWLEDGEMENTS

The journey to where I am now was long and had many curves and obstacles. I would not be able to make it without help from many people.

First and foremost, I would like to express my deepest gratitude to Prof. John Verboncoeur for the opportunity to pursue a Ph.D. at Michigan State University. Your endless knowledge and advice I received during my study helped me better understand plasma physics. I am forever in debt for your patient and the possibility I had to find my way, although it (usually) took longer to reach the destination than expected.

I would like to thank other committee members, Dr. Michael Murillo, Dr. Brian O'Shea, Dr. Mohsen Zayernouri, and Dr. Peng Zhang. I gain valuable experience and knowledge through lectures and casual discussions. I admire the passion for the research and teaching you all have, and I hope to have it in years to come.

I would like to offer my special thanks to the Plasma Theory and Simulation Group (PTSG) members, Prof. Yangyng Fu and Dr. De-Qi Wen, for their support with writing, simulations, and plasma and research discussions. I would like to extend my gratitude to Dr. Guy Parsey for conversations on numerous topics and entrusting me with the development of the KGMf.

My journey to graduate school at Michigan State University started much earlier. I would like to express my gratitude to Assoc. Prof. Leon Kos and Prof. Jože Duhovnik for guiding me through my undergraduate and master's studies. I will be forever in debt to Dr. Nikola Jelić for introducing me to a beautiful world of plasma physics, patiently guiding me through my first steps in plasma simulations, and giving me a push when a push was needed.

Last but not least, I would like to thank my parents for their unlimited support and understanding with all my life decisions, among which pursuing the Ph.D. was probably one of the boldest. Without your support, I would never make it. Thank you!

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	x
CHAPTER 1 INTRODUCTION	1
1.1 Modelling of Low-temperature Plasmas	2
1.1.1 Kinetic Models	5
1.1.2 Fluid Models	8
1.1.3 Hybrid Models	10
1.1.4 Global Models	10
1.2 The Motivation	12
1.2.1 Coupling of Self-Consistent Evaluation of the EEDF with a Global Model	12
1.2.2 Sensitivity Analysis and Global Models	13
1.2.3 Prioritizing Reaction and Species	14
1.3 Outline of the Thesis	16
CHAPTER 2 THE KINETIC GLOBAL MODEL FRAMEWORK (KGMF)	17
2.1 Introduction	17
2.2 The Main Governing Equations	18
2.3 Boltzmann Equation and Electron Energy Distribution Function (EEDF) . .	20
2.3.1 Shape Factor x for Analytical Rate Coefficients K_i	23
2.3.2 Boltzmann Equation Solvers	24
2.4 Coupling the KGMf with Boltzmann Equation Solvers	25
2.4.1 EEDF Evaluation Frequency	28
2.4.2 Coupled BE Solvers	29
2.5 Benchmark of the KGMf with Coupled Boltzmann Equation Solver	30
2.5.1 Temporal Evolution of Plasma Parameters	32
2.5.2 Computational Implication of Including a BE Solver	36
2.5.3 Steady-state Densities	37
2.5.4 High Power Microwave Breakdown	39
CHAPTER 3 GLOBAL SENSITIVITY ANALYSIS	42
3.1 Uncertainty Quantification and Sensitivity Analysis	42
3.2 The Elementary Effects Method	44
3.3 Variance-based Methods	46
3.3.1 Sensitivity Indices	47
3.3.2 Computing Sensitivity Indices	49
3.4 Global Sensitivity Analysis in Plasma Discharges	51
3.4.1 Uncertainties of Reaction Rate Coefficients	52
3.4.2 Plasma Discharge in Argon	54

3.4.2.1	GSA Parameters	56
3.4.2.2	GSA Results	56
3.4.2.3	Conclusion	61
3.4.3	Plasma Discharge in H ₂ -O ₂ -Ar Mixture	62
3.4.3.1	GSA Parameters	63
3.4.3.2	GSA Results - output distributions	64
3.4.3.3	GSA Results - reaction importance	70
3.4.3.4	Conclusion	72
3.5	Conclusion	73
CHAPTER 4 PRIORITIZING REACTIONS AND SPECIES		75
4.1	Introduction	75
4.2	Methods for Ranking Species and Reactions	77
4.2.1	Reaction Pathways	80
4.2.2	Directed Relation Graph (DRG)	81
4.2.2.1	Creating Graph and Skeleton Mechanism	84
4.2.3	Error Between Detailed and Reduced Reaction Sets	87
4.2.4	Reaction Set Reduction Process	89
4.3	Plasma Discharge in H ₂ -O ₂ -Ar Gas Mixture	91
4.3.1	Capturing H density	93
4.3.2	Capturing H ₂ ⁺ density	97
4.3.3	Conclusion	102
4.4	Conclusion	104
CHAPTER 5 CONCLUDING REMARKS AND FUTURE WORK		106
5.1	Future Work	108
5.1.1	The KGMf and EEDF Solvers	108
5.1.2	Chemistry (Reaction) Kinetic Engine	109
5.1.3	Interconnected KGMf Codes	110
5.1.4	Energy Dependence in Reaction Set Reduction	111
5.1.5	Machine Learning for Defining Input Parameters in Sensitivity Analysis	112
APPENDICES		113
APPENDIX A	Models added to the KGMf	114
APPENDIX B	Computational Additions to the KGMf	117
APPENDIX C	Reaction Set for Global Sensitivity Analysis	143
BIBLIOGRAPHY		148

LIST OF TABLES

Table 2.1	The reaction set for the benchmark case. The reaction set considers the following species: ground state atoms (Ar), electrons (e), excited state atoms (Ar [*]), atomic ions (Ar ⁺), and molecular ions (Ar ₂ ⁺). <i>Notes:</i> T_e in eV and T_{gas} in K.	31
Table 2.2	The impact of different changes of the reduced electric field E/N (used to define the EEDF evaluation frequency in the KGMf) on total computational times and steady-state densities. Total running times for the KGMf (t_K) and ZDPlasKin (t_Z) are given in seconds. The percent difference between the KGMf and ZDPlasKin is given for the following species densities: electrons (e), excited state atoms (Ar [*]), atomic ions (Ar ⁺), and molecular ions (Ar ₂ ⁺).	35
Table 3.1	Reactions used in the plasma discharge in argon with their reaction rate coefficients and threshold energies. <i>Note:</i> T_e in eV, T_g in K.	55
Table 3.2	Species used in the reaction set for 250 ns plasma discharge in H ₂ -O ₂ -Ar gas mixture with 1% H ₂ , 0.15% O ₂ , and 98.85% Ar. The reaction set is presented in Tab. C.1 (section C).	62
Table 3.3	The most important reactions for electron density at the pulse peak (time between 50.24 and 50.5 μ s) according to the total sensitivity index S_{T_i}	72
Table 4.1	Reaction rate threshold, number of species and reactions, and error in H density from DRG method in each reduction step for two types of EEDFs: Maxwellian EEDF (top) and EEDF from BS solver (bottom). The error was computed between simulation results from detailed and reduced reaction set for the time intervals shown in Fig. 4.3. The last column presents simulation time reduction in the KGMf with a given reaction set compared to simulations with a detailed reaction set. <i>*Note:</i> For reaction set reductions, forward and backward reactions were treated separately, thus making the total number of reactions 215, the KGMf reaction set has 146 reactions.	94

Table 4.2	Reaction rate threshold, number of species and reactions, and error in H_2^+ density from the DRG method in each reduction step for reductions with Maxwellian EEDF and EEDF from the BE solver. The error was computed between simulation results from detailed and reduced reaction sets in time intervals shown in Fig. 4.3. The last column presents simulation time reduction in the KGMf with a given reaction set compared to simulations with a detailed reaction set. <i>*Note:</i> For reaction set reductions, forward and backward reactions were treated separately, thus making the total number of reactions 215, while the KGMf reaction set has 146 reactions.	99
Table B.1	Dimensions and their sizes when storing collected results in sensitivity analysis example presented in listing B.5.	130
Table C.1	Reactions used in the plasma discharge in argon with their reaction rate coefficients and threshold energies. Additional subscripts v , r , and e indicate vibrational, rotational and excited states, respectively. <i>Notes:</i> T_e is the electron temperature, in eV, T_{gas} is the gas temperature, in K.	143

LIST OF FIGURES

Figure 1.1	Schematic representation of multi-scale modeling in terms of time and space scales.	4
Figure 1.2	Main loop of Particle-in-cell (PIC) method with included collisions. . . .	6
Figure 2.1	Plots of different EEDF shapes: from simulations (EEDF from BE) and from Eq. (2.13) with different shape factors x	24
Figure 2.2	Flowchart of the KGMf and BE solver coupling via <code>EEDFsolver</code> class. The <code>EEDFsolver</code> class is initialized according to selected simulation parameters. In the time advancement of the ODE system, parameters are set in the <code>EEDFsolver</code> class, and the decision is made whether to call the selected BE solver to evaluate the EEDF or use the previously computed EEDF to compute reaction rate coefficients.	26
Figure 2.3	Temporal evolution of the reduced electric field E/N and the effective electron temperature T_e . Pressure $p = 760$ Torr in argon and absorbed power density $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$	32
Figure 2.4	Temporal evolution of the EEDFs from BOLOS (KGMf), BOLSIG+ (ZDPlasKin) and analytical Maxwellian EEDF (KGMf) for the same effective electron temperature (T_e) and at the same times. Maxwellian EEDFs (dotted lines) show higher probability of electrons with energies above ~ 20 eV compared to EEDFs from BOLOS and BOLSIG+ at each presented time. The pressure $p = 760$ Torr and the absorbed power density $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$	33
Figure 2.5	Time dependence of species densities: (a) electron and excited species, (b) the atomic and molecular ion densities. Steady-state background gas density $n_g = 2.44 \times 10^{19} \text{ cm}^{-3}$ and the ionization degree $\chi = 1.05 \times 10^{-8}$. The pressure is $p = 760$ Torr and the absorbed power density is $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$	34
Figure 2.6	Time evolution of reaction rates for (a) reactions defined by cross sections and (b)-(d) reactions defined in Arrhenius form. Reaction labels are from Table 2.1. The pressure is $p = 760$ Torr and the absorbed power density is $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$	35
Figure 2.7	The steady-state results for fixed deposited power density ($\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$) and a range of gas pressures ($p = 50 - 1000$ Torr).	38

Figure 2.8	The steady-state results for fixed pressure ($p = 760$ Torr) and a range of deposited power densities ($\tilde{P}_{\text{abs}} = 10 - 1000 \text{ W}\cdot\text{cm}^{-3}$).	39
Figure 2.9	Comparison of breakdown times in high power microwave argon discharges for PIC simulation and global models with EEDF defined analytically ($x = 1$ and $x = 6.5$) and BE solvers (BOLOS and BOLSIG+). Analytically defined EEDFs were constant in time, while EEDFs evaluated by the BE solvers were time-dependent. The electric field amplitude $E_0 = 2.82 \text{ MV/m}$ and the angular frequency $\omega = 2.85 \text{ GHz}$. The PIC results were taken from Ref. [37].	41
Figure 3.1	Representation of 2D grid with two parameters ($p = 2$), four grid points ($k = 4$), and parameter change $\Delta = 2/3$, used for defining parameter values for the elementary effects method. Each arrow presents points needed to compute one elementary effect: horizontal arrow to identify elementary effect for x_1 , vertical arrow for x_2	45
Figure 3.2	Scatter plot of model realizations (model output values) depending on values of one parameter, a multiplication factor F_u for reaction R110. Points along vertical lines have the same value for factor F_u , but different values for other factors. The use of factor F_u is defined in section 3.4.1.	48
Figure 3.3	The comparison of distributions for sampling factors for reaction rate coefficient factors: log-normal distribution samples and the distribution (bar charts and solid line, respectively), and normal distribution (dotted line) with mode (the most probably value) of 1.0 and standard deviation $\sigma = 0.2$. The interval that encloses 99% of the probability of log-normal distribution is labeled with its low (0.64) and high (1.71) boundary. The most probable value of the coefficient is shown as vertical dashed line.	53
Figure 3.4	The initial distributions of reaction rate coefficient factors for (a) elastic scattering (reaction R1) and (b) ionization (reaction R2) with $N = 100$ samples. The dashed line shows the log-normal distribution, and the histogram shows the distribution of samples. The vertical dash-dotted line shows the distribution mode (the most probable value of the distribution).	56
Figure 3.5	Time evolution of densities (a) and electron temperature (b) in the simulation case (base case) without uncertainties in the reaction rate coefficients. Results acquired in the last 200 time steps of the simulation (shaded area on the plot) were used in a global sensitivity analysis.	57
Figure 3.6	Comparison of density mode and intervals that are enclosing 99% of all densities of given species. Reaction rate coefficient distributions had mode 1.0, $\sigma = 0.2$, and were sampled with $N = 100$ samples.	57

Figure 3.7	The distribution of species densities (results of the simulation) of selected species: (a) molecular ions Ar_2^+ , (b) electrons, (c) atomic ions Ar^+ , and (d) excited species Ar^* . Densities from results are shown as histograms. The distributions fitted to results and the input distributions are shown as dashed lines. The vertical dotted line marks the most probable values from each distribution.	58
Figure 3.8	Relative importance of reactions on species' densities for (a) molecular ions Ar_2^+ and (b) electrons. The relative importance was defined according to the sensitivity indices S_{T_i} and S_i	60
Figure 3.9	The relative importance of reactions on species' densities for (a) atomic ions Ar^+ and (b) excited species Ar^* . The relative importance was defined according to sensitivity indices S_{T_i} and S_i	60
Figure 3.10	The shape of the deposited power in pulse discharge in 1% H_2 , 0.15% O_2 , and 98.85% Ar gas mixture. The pulse was taken from experiments shown in Ref. [110] with the repetition frequency $\nu = 20\text{kHz}$ ($\tau = 50\mu\text{s}$) and pulse duration 250 ns. The shape of the pulse follows an experimentally measured power profile, but taking only the positive part of the pulse and depositing the same amount of energy in every pulse as in the experiment (2.6mJ/pulse).	63
Figure 3.11	The histogram of $N = 100$ samples drawn from the log-normal distribution with the dashed line showing the initial log-normal distribution. The figure shows the multiplication factor distribution for reaction 110: $\text{Ar}^+ + \text{H}_2 \rightarrow \text{Ar} + \text{H}_2^+$	64
Figure 3.12	Time evolution of selected species in 250 ns pulse discharge in $\text{H}_2\text{-O}_2\text{-Ar}$ gas mixture. The light gray shaded areas mark the time intervals at which values for GSA were recorded. The GSA was made at two time intervals: (a) at the peak of the pulse (time between 50.24 and 50.5 μs), and (b) at the plateau before the next pulse (between 98 and 100 μs).	65
Figure 3.13	Comparison of density distribution mode and intervals that are enclosing 99% of all density values for selected species at (a) the peak of the pulse (time between 50.24 and 50.5 μs), and (b) the plateau before the next pulse (between 98 and 100 μs). All reaction rate coefficient distributions had mode=1.0 and $\sigma = 0.2$	66

Figure 3.14	The distributions of species densities (results of the simulation) of selected species: (a) at pulse peak and (b) at relaxation time for the atomic ions Ar_2^+ , and (c) at pulse peak and (d) at the relaxation time for the molecular hydrogen ions H_2^+ . The species densities are shown as histograms, the log-normal distributions fitted to the results, and the input distributions are shown as dashed lines.	67
Figure 3.15	The distribution of electrons density at (a) the peak of the pulse (time between 50.24 and 50.5 μs), and (b) the plateau before the next pulse (between 98 and 100 μs). The density values are shown as histograms and the distributions as dashed lines.	68
Figure 3.16	The distribution of electron temperature T_e at (a) the peak of the pulse (time between 50.24 and 50.5 μs), and (b) the plateau before the next pulse (between 98 and 100 μs). The temperature values are shown as histograms and log-normal distributions as dashed lines.	69
Figure 3.17	The relative importance of reactions on H_2^+ density at (a) the peak of the pulse (time between 50.24 and 50.5 μs), and (b) the plateau before the next pulse (between 98 and 100 μs). The relative importance was defined with sensitivity indices S_{T_i} and S_i	70
Figure 3.18	The relative importance of reactions on electron density at the pulse peak (time between 50.24 and 50.5 μs). The relative importance was defined with sensitivity indices S_{T_i} and S_i	71
Figure 4.1	The number of species and reactions in the reduced reaction set depending on the relative reaction contribution threshold (ε) on the atomic hydrogen density (H). The relative contributions were computed at multiple times-of-interest (see Fig. 4.3).	82
Figure 4.2	Example of a graph presenting a detailed reaction set for an $\text{H}_2\text{-O}_2\text{-Ar}$ gas mixture (55 species and 215 reactions, detailed reaction set in Table C.1). The target species, H (node labeled as "H_g"), is positioned in the center of the graph. Species in the reaction set are nodes, and reactions are edges (connections). The number on each edge is the reaction number. Forward and backward reactions are separated and have unique numbers.	84

Figure 4.3	Time evolution of species densities (electrons, H, and H_2^+) and selected times of interest. The error in quantities of interest (QoI) from a detailed and reduced reaction set is computed at selected times of interest, marked with red vertical dashed lines. Multiple times of interest add to the complexity of the reduction process but decrease the error from reduced reaction set simulations in cases when selected QoI changes in time.	88
Figure 4.4	A flowchart of reaction set reduction with the KGMf. In the top row are input parameters for reaction set reduction, and the center part shows the loop required to reduce the reaction set until the given maximum error threshold is reached. <i>Note:</i> 'RS' stands for 'reaction set'.	91
Figure 4.5	The number of species and reactions vs. relative net contribution of reactions (ε) on H density with Maxwellian EEDF (a) and EEDF from coupled Boltzmann equation solver (b). <i>Note:</i> For reaction set reductions, forward and backward reactions are counted separately, making the total number of reactions 215, while the KGMf reaction set has 146 reactions.	93
Figure 4.6	Time evolution of H density from simulations with Maxwellian EEDF and EEDF from the BE solver. The assumption of Maxwellian EEDF causes up to 32 % error in H density, which could be acceptable for some applications. Vertical red dashed lines mark the start of each pulse.	95
Figure 4.7	Time evolution of H density from detailed and reduced reaction sets using Maxwellian EEDF (a) and EEDF from the BE solver (b). Vertical red dashed lines mark the start of each pulse.	96
Figure 4.8	Time evolution of relative error in H densities introduced by reduced reaction sets with Maxwellian EEDF (a) and EEDF from BE solver (b). Relative errors introduced by reduced reaction sets are smaller with each additional reduction run. Hydrogen atom density at times-of-interest (Fig. 4.3) were used to compute errors in the DRG method (δ). The error in the DRG method is added to the legend for each reduction step and is very close to the relative error in H species densities in the left figure.	97
Figure 4.9	Dependence of the number of species and reactions on the relative net contribution of reactions on H_2^+ density using Maxwellian EEDF (a) and EEDF from the BE solver (b). <i>Note:</i> For reaction set reductions, forward and backward reactions are counted separately, making the total number of reactions 215, while the KGMf reaction set has 146 reactions.	98

Figure 4.10	Time evolution of H_2^+ density from simulations with Maxwellian EEDF and EEDF from the BE solver. The assumption of Maxwellian EEDF causes up to 8 times higher densities (green dashed line). Vertical red dashed lines mark the start of each pulse.	100
Figure 4.11	Time evolution of H_2^+ density from detailed and reduced reaction sets using Maxwellian EEDF (a) and EEDF from the BE solver (b). The H_2^+ density follows the electron density during a single pulse. For readability reasons, only selected reduction steps are shown. Vertical red dashed lines mark the start of each pulse.	100
Figure 4.12	Time evolution of relative error in H_2^+ densities introduced by reduced reaction sets with Maxwellian EEDF (a) and EEDF from the BE solver (b). The H_2^+ density follows the electron density during a single pulse. Errors computed in the DRG method (δ) are displayed in the legend for each reduction step. The plot of relative errors (b) shows that errors introduced by reduced reaction sets could be orders of magnitude different from the supplied reaction threshold ε , meaning there is no direct connection between the two values. For readability reasons, only selected reduction steps are shown. Vertical red dashed lines mark the start of each pulse.	101
Figure B.1	Verbose output of running all tests.	119

CHAPTER 1

INTRODUCTION

Plasma, sometimes called the fourth state of matter, exists in many places (99 % of visible matter) in either natural or manufactured form. In nature, the most recognizable forms of plasma are lightning, stars, and galaxies. Artificially created plasma is used in many terrestrial applications, from material processing (welding, cutting, etching, deposition, fabrication of integrated circuits), light sources (fluorescent lamps), disinfection and wound treatment in health care, and treatment of fresh fruits to withstand mold [1, 2].

Plasma is a gas with charged particles that give plasmas unique properties when put under the influence of an electromagnetic field. Charged particles with different charge polarities, positive (ions), negative (electrons, ions), and neutrals (atoms, molecules), react differently to applied electric and magnetic fields. This property leads to interesting phenomena in plasmas, such as transport, creation, and destruction of particles [3]. Phenomena in LTPs, due to the complicated reaction kinetics, may vary across a wide range of time scales, from \sim ns time scale for ionization wave propagation to \sim ms time scale for multi-phase chemical reactions and ignition delays.

Low-temperature plasmas (LTP) are composed of many species, including neutral atoms and molecules, excited state particles, radicals, ions, and electrons. They are weakly ionized (usually from a fraction of a percent up to tens of percent) and far from thermal equilibrium. The electron temperature (T_e) is much higher than temperatures of heavy charged particles (T_i) and the background gas (T_g): $T_e \gg T_i \simeq T_g$. The characteristic electron temperature is a few eV to 10 eV (1eV = 11905 K), and temperatures of both heavy particles and the background gas are usually up to 1000 K. Energetic electrons have enough energy to initiate reactions that can create radicals, charged and excited state particles, and photons, making LTPs a very chemically rich environment even at room temperatures. This property makes

LTPs a good fit for processing media in many fields, from the fabrication of microprocessors to plasma medicine and nanomaterial synthesis. The versatility of the LTPs is driving its research and keeps the field of research very active, but many fundamental questions remain unresolved [1].

1.1 Modelling of Low-temperature Plasmas

Fundamental interactions in plasmas are very well defined and researched. However, the sheer number of possible interactions between charged particles, background gas, and materials makes plasma research very complicated. Analytical description of fundamental processes in plasmas is possible but very complicated and cumbersome for engineering use. Introduction of models, a simplified description of the physical system, is needed to describe plasma at the engineering level. The model has to adequately describe the main characteristics of the system by capturing the time evolutions of system processes and parameters, e.g., how the system reacts to an applied external electric field, the time evolution of species density, or change in pressure due to increased temperature. Therefore, selecting a suitable model for the physics system is essential. It is possible to have multiple models describing certain phenomena, depending on the focus of the simulation, e.g., modeling of a welding arc can have a model for simulating the temperature of the arc (needed for a weld to be formed) or a model for the chemical composition of the arc (mixture of material from welding wire, protective gas, and base material).

Various mathematical and numerical techniques can be used to describe models, and they present a tool to materialize those models in the form of algorithms and simulation codes. Algorithms and their implementations as simulation codes inherently include assumptions and uncertainties that might render the model inappropriate for a selected system. Different models can be developed for describing the same phenomena, and models can have different implementations into the simulation code, i.e., using different algorithms or numerical imple-

mentation. With all that said, it is imperative to assess the discrepancy in results between the model and simulation codes (verification) as well as between the model and the experiments (validation). The results from experiments are considered correct, although they can natively include various uncertainties, e.g., measurements or inappropriate experiment setup. The assessment is done in two steps [4, 5]:

- **Verification** is where implementation of the model is tested against the theory; here, the check is focused on whether the code returns the same result as the theory predicts.
- **Validation** is comparison of the simulation results to experimental results.

Verification and validation are equally important as they assess different parts of the model implementation, and to develop high-fidelity simulation codes, both have to be performed. Another type of test is benchmarking, where the same input parameters are used in two or more different simulation codes, and the results from codes are compared.

Phenomena in plasmas can vary by many orders of magnitude in time and space scales, and this requires different models to capture important processes in the simulated system (Fig. 1.1). It is necessary to choose an appropriate model (and the simulation code implementing the selected model) to capture the physics of the phenomena while reasonably limiting the simulation time. Models accurately describe key phenomena at selected time and spatial scales and utilize approximations and assumptions to evaluate the neglected or missing quantities, i.e., quantities on other time and spatial scales.

Simulation codes use independent spatial and temporal variables, combined with models, to compute variables of interest, e.g., pressures, temperatures, and velocities in the modeled system. They can be classified according to the number of spatial coordinates they support - 0D (global models), 1D, 2D, and 3D codes. A higher number of spatial dimensions generally means higher code complexity due to the complexity of underlying equations in the model. Reduction of dimensionality (from 3D to 2D or 1D) is often possible because the system of interest is symmetric, or changes of system parameters are (almost) negligible in one

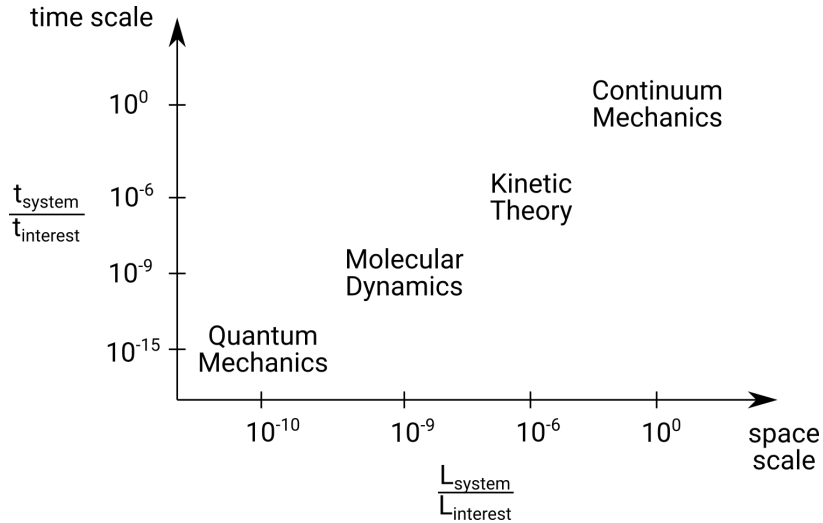


Figure 1.1 Schematic representation of multi-scale modeling in terms of time and space scales.

(or two) dimensions. The reduction of dimensionality is often desirable because it reduces the complexity of the simulation and results in a shorter simulation time. The reduction of dimensionality also enables including additional aspects of the system into the model. For example, the lower-dimensional model allows computing more complicated domain geometries (e.g., free-shape domains or additional geometrical features, like protrusions), more complicated velocity distributions for particle species, and more complicated or detailed reaction sets (e.g., a higher number of species and more reactions). Multi-scale kinetic modeling of plasma chemistry in LTPs is computationally expensive, and global (volume-averaged) models are commonly employed to understand the reactions and limit the scope of multi-dimensional models [6, 7]. Reaction kinetics of the LTPs are crucial for many plasma applications, such as material processing [8] and plasma medicine [9, 10]. The versatility of LTPs drives its research to resolve many fundamental questions [11]. Global models can be used when the spatial variation of system variables is small or negligible and when seeking general trends of a target system parameter (e.g., absorbed power and gas pressure).

1.1.1 Kinetic Models

The main focus of kinetic models is describing the movement of particles in the system as a response to the electric and magnetic fields, collisions between particles, and interactions with boundaries (e.g., reflections and scattering). This can be achieved by solving the Vlasov-Maxwell (or Vlasov-Poisson for electrostatic case) system of equations, where particle distribution function $f(\vec{r}, \vec{v}, t)$ defines the number of particles at spatial location \vec{r} , having a velocity \vec{v} at time t , and the numerical solution to the equations have very few approximations [12].

Depending on time and spatial scales, kinetic modeling encompasses different methods. Molecular dynamics (MD) focuses on interactions between individual particles, including forces from various potentials and collision models. MD models are grid-less methods, and the forces between particles are computed directly. Direct computation of forces on particles results in $O(N^2)$ computational complexity for long-range interactions, limiting the number of particles and the size of the simulated system. The Tree-code method, introduced by Barnes and Hut [13] in 1986 for computing forces in stellar dynamic applications, excels in systems with nonuniform particle density. The forces between particles are computed either directly particle-to-particle or particle-to-cluster, where the clusters are dynamically created to present the effect of many particles. The next step in grouping particles into clusters is the particle-in-cell (PIC) method. Individual particles with similar properties, i.e., position, mass, and velocity, are grouped into super-particles (also called computer particles) and treated as single particles in the simulation. The method was formalized in the 1970s by Birdsall and Langdon [14] and Hockney and Eastwood [15]. It is widely used for modeling processes on μs and μm spatial and time scales. Forces between particles are computed via the potential on a grid and currents between grid points if the magnetic field is considered. Figure 1.2 presents the main loop in a PIC code. The particle charges are mapped to grid points, where the Poisson equation for the electric field is solved, and forces on particles are

computed from the electric field. The particle's velocity and positions are computed from the force acting on the particle due to electric and magnetic fields. The collisions are included in the equations via Monte Carlo collisions, i.e., random collisions between particles based on probabilities of particle interactions, and are considered in the force acting on each particle (\vec{F}_i in Eq. (1.1)).

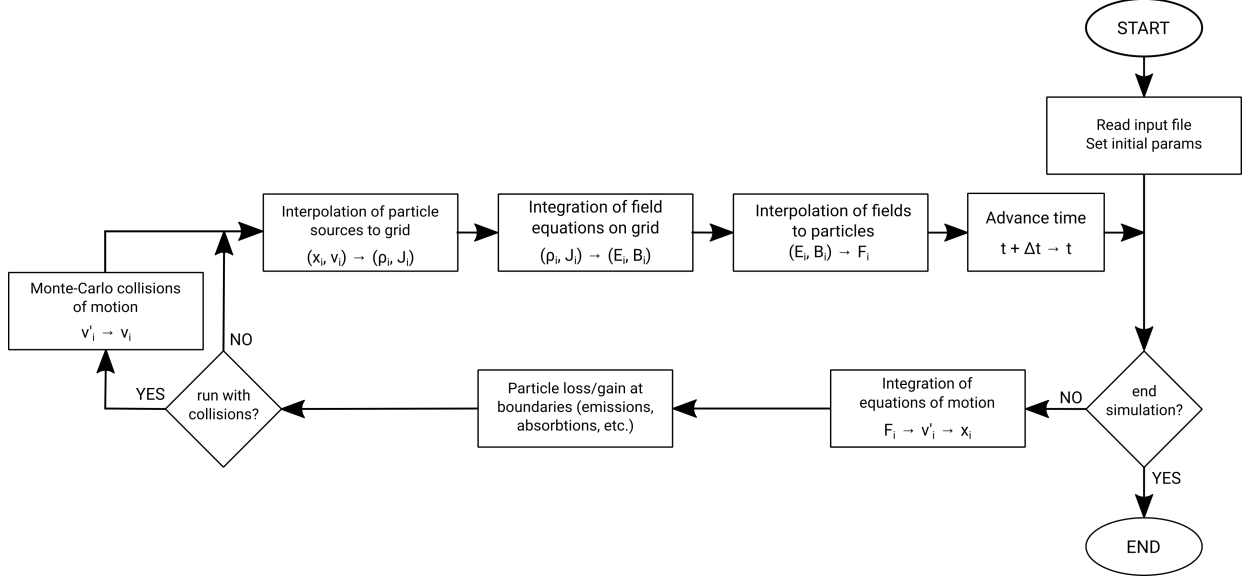


Figure 1.2 Main loop of Particle-in-cell (PIC) method with included collisions.

The computational complexity of the tree-code and the PIC methods is $O(N \log N)$, enabling modeling of larger systems in terms of the number of "real" particles and spatial dimensions. Each super-particle can represent between $10^5 - 10^9$ real particles, and the ratio of real to super-particles can vary during the simulation. The tree-code and the PIC governing equations describe particles' position and velocity in time and can be written using the leap-frog method:

$$\begin{aligned} \vec{v}_i(t + \Delta t/2) &= \vec{v}_i(t - \Delta t/2) + \frac{\vec{F}_i(t)}{m} \\ \vec{r}_i(t + \Delta t) &= \vec{r}_i(t) + \frac{\vec{v}_i(t + \Delta t/2)}{\Delta t} \end{aligned} \quad (1.1)$$

where i is the particle index running over all particles in the system, \vec{r}_i is the position of the particle, \vec{v}_i is the velocity of particles, \vec{F}_i is the sum of all forces on the particle, and

Δt is the time step. Thus, the particle positions and velocities are computed at different times, the positions are computed at integer time steps (t), and velocities are computed at half-time steps ($t + \Delta t/2$). This time splitting results in second-order method [14, 16].

The PIC method is based on first principles and does not include many approximations but has limitations in terms of spatial and temporal steps to keep the method stable. The grid steps Δx and time steps Δt have to be smaller than the smallest characteristic value to adequately describe processes in the system, e.g., sheath length or Debye length for Δx and plasma or collision frequency for Δt . An additional limitation is a necessary condition for convergence of the numerical method used to solve PDEs, imposed by the Courant–Friedrichs–Lewy condition: $c\Delta t < \Delta x$, where c is the speed of light in vacuum. Knowing particle positions and velocities enables computing various plasma parameters, such as particle densities, density and energy fluxes in space and time, and energy distribution functions (EDFs), with high fidelity. The electron energy distribution function (EEDF) can be obtained from PIC methods. It can be used in other simulation methods, e.g., fluid or volume-averaged (0D or global) models that depend on the EEDF. The non-equilibrium properties of the simulated systems can be described accurately. While the computational complexity of the method increases with the number of particles, the PIC method is suitable for low-temperature plasmas. The main properties can often be captured with a smaller number of particles.

The usability of the PIC method can be observed in the abundance of available PIC codes, ranging from open-source to commercial codes. The minimal setup for the PIC method is relatively simple; adding various boundary conditions, complex geometries, and collision operators increase the code’s complexity. The Plasma Theory and Simulation Group (PTSG) developed a suite of PIC codes in C/C++ languages for 1D and 2D systems in all types of coordinate systems (Cartesian, cylindrical, and spherical): XES1, XPDP1, XPDP2, XPDC1, XPDC2, XPDS1, oopd1, XOOPIIC [14, 17]. The oopd1 and XOOPIIC codes are object-oriented codes that abstract the parts of simulation (plasma device, boundaries, fields,

reactions) to increase the flexibility of adding new functionality and physics models. A wide variety of other PIC codes are available, such as the open-source codes ALaDyn [18] and WarpX [19, 20], and the commercial codes VSim [21], Comsol [22], MAGIC [23], and ICEPIC [24].

1.1.2 Fluid Models

Fluid models take the opposite approach to the kinetic models and focus on defining an Eulerian distribution of plasma in the system. The particles in the system are described through their collective effects, employing the transport of quantities in the systems to describe their motion, i.e., transport of density, momentum, and energy in the systems. The trajectories and velocities of individual particles are not known and cannot be recovered using fluid models. By focusing on the collective effects of particles by treating them as fluids, the simulations can describe systems with larger time and spatial scales [12]. Fluid models have more similarities with experiments than kinetic models. Their input parameters and results are often quantities measured in experiments, i.e., the energy deposited to the system, the temperature of the outflow gas mixture, and pressure.

A Boltzmann transport equation (BE) is one of the most general descriptions of non-equilibrium fluid, defining the probability of a typical particle to be at a given small volume and have a given velocity [25]:

$$\mathbb{B} = \frac{\partial f}{\partial t} + \vec{v} \cdot \nabla f + \vec{a} \cdot \nabla_v f = \left(\frac{\delta f}{\delta t} \right)_{coll} \quad (1.2)$$

where $f = f(t, \vec{x}, \vec{v})$ is the particle distribution function for species, \vec{v} is the velocity and \vec{a} is the acceleration of the fluid element, ∇ and ∇_v are del operators acting in position and velocity space, respectively. The term on the right hand side of the equation, $\left(\frac{\delta f}{\delta t} \right)_{coll}$, presents the time rate of change in distribution function f due to the collisions.

Relevant macroscopic quantities, such as continuity and energy transport equations, can be derived from Eq. (1.2) in the form of spatial and temporal differential equations. For

example, the continuity equation for each species in the system is derived by taking the zeroth velocity moment of Boltzmann equation, $\int_v (\mathbb{B}) d^3v$, and is the following:

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\vec{u}) = \frac{S_m}{m} = \int_v \left(\frac{\delta f}{\delta t} \right)_{coll} d^3v \quad (1.3)$$

where n is the particle density, $\vec{u} = \frac{1}{n} \int \vec{v} f(v) d^3v$ is the average velocity, and S_m is the source term for particles. The source term includes volumetric and surface processes involving species, e.g., creation and destruction of particles due to collisions with other species, and surface recombination.

The moment transport equation is derived from Boltzmann transport equation by multiplying it by particle mass $m\vec{v}$ and integrating over velocity space, $m \int_v (\vec{v} \mathbb{B}) d^3v$, yielding the following equation [25]:

$$n \frac{\partial \vec{u}}{\partial t} + n(\vec{u} \cdot \nabla) \vec{u} + \frac{1}{m} \nabla \cdot \vec{\vec{P}} - \frac{nq}{m} (\vec{E} + \vec{u} \times \vec{B}) = \frac{\vec{S}_p}{m} = \int_v (\vec{v} - \vec{u}) \left(\frac{\delta f}{\delta t} \right)_{coll} d^3v \quad (1.4)$$

where $\vec{\vec{P}}$ is the pressure tensor, and \vec{E} and \vec{B} are the electric and magnetic fields, respectively. The pressure tensor $\vec{\vec{P}}$ takes into account scalar pressure and tangential shear forces (viscus forces). In a system with weak or zero magnetic field, when viscus forces can be neglected [25], the nondiagonal terms in $\vec{\vec{P}}$ are zero, and $\nabla \cdot \vec{\vec{P}}$ can be written as $\nabla \cdot \vec{\vec{P}} = \nabla p$, where p is the static pressure.

The fluid equations present a non-closed set of equations, i.e., the equation set cannot be solved self-consistently. If the first two levels are considered, i.e., species continuity and momentum transfer, then the pressure tensor $\vec{\vec{P}}$ has to be specified or computed from the third moment of the Boltzmann equation. The third velocity moment of the Boltzmann equation is the heat transfer equation that brings another unknown into the system of equations. To stop this circle of a not self-consistent system of equations, the missing terms on a selected level are either approximated for the individual system or derived from kinetic models, i.e., electron energy distribution functions, average velocities, energy transfers.

There are an abundance of fluid simulation codes available under open-source and commercial licenses. Open-source codes are BOUT++ [26, 27] for particle and heat fluxes in

Edge Localised Modes (ELMs) in tokamaks, and SOMAFOAM [28], an OpenFOAM based framework for performing continuum simulations of low-temperature plasma. The commercial software packages include TechX's USim [29], COMSOL Multiphysics [22], ESI's CFD-ACE+ Suite [30], and Plasma Matters' PLASIMO [31].

1.1.3 Hybrid Models

Hybrid models combine the advantages of kinetic and fluid models: taking advantage of the fidelity of the kinetic model and combining it with the computational speed of the fluid model. Each part of the hybrid code, i.e., kinetic and fluid part, is focusing on capturing the physics at different time and spatial scales in the system [11, 12]. One common use of hybrid codes is to use the kinetic model for capturing the motion and interaction of electron particles due to their fast nature and the fluid model for capturing other species that are usually heavier and slower. Since both models focus on different time and spatial scales, the communication and exchange of computed values between models are of the essence.

It may look like using two different models will increase the complexity of the simulations to the point of making them intractable, but hybrid simulation codes are widely used. PIC-hybrid codes are treating electrons as particles and other species as fluid [32]. Hybrid codes also enable users to treat important species (i.e., species of interest) as particles to capture their evolution in detail and the rest of species as fluids [33]. Hybrid codes are utilized in various simulations, e.g., HPEM [11] for low-temperature plasma reactors, and Pegasus [34] for astrophysical plasma dynamics.

1.1.4 Global Models

Global (0D or volume-averaged) models are used to focus the research on describing the volume-averaged parameters in plasma systems and to explore complicated plasma chemistry and reaction kinetics [35]. Global models do not provide nor request any information

about the spatial distribution of system variables. All variables, either independent (input) or dependent (output), are spatially invariant, but they can include spatial (non-local) effects (e.g. effective temperature, Eq. (1.5)). Reduced dimensional complexity of underlying equations enables the expansion of the model by including more complex chemistry, reaction kinetics and dynamics, and different energy sinks or sources. Global models can be used when the spatial variation of system variables is small or negligible in the interest of finding a general trend of a target system parameter (e.g., temperature, pressure). Global models are also a great tool to learn the general behavior of the system for a certain set of input parameters and validate or predict experimental measurements. Considering their short run times and usually small memory footprint, doing parameter sweeps with multiple runs in parallel is possible on a desktop computer.

The electron temperature can be computed from the energy equation as part of the ODE system in the global model or from supplied or evaluated EEDF using the following equation:

$$T_{\text{eff}} = \frac{2}{3} \langle \varepsilon \rangle = \frac{2}{3} \int_0^\infty \varepsilon f_0(\varepsilon) d\varepsilon \quad (1.5)$$

where T_{eff} is the effective electron energy in eV, $\langle \varepsilon \rangle$ is the mean electron energy in eV, and $f_0(\varepsilon)$ is the normalized EEDF. The EEDF is required in global models and can be predefined, either in form of tables for different values of reduced electric field E/N or as an analytical expression of a shape factor x (Eq. (2.13) in Section 2.3.1), or is computed by evaluating the Boltzmann equation.

Global models are great tools for learning the general behavior of the system for a given set of input parameters and to validate or predict experimental measurements, e.g., plasma-assisted combustion [36], high-power microwave breakdown [37, 38], jet flames [39], coupled reaction chains [40], and the validity of LTP scaling laws [41, 42]. Many global model simulation codes are available, either as commercial codes that can run as global models (COMSOL [22]) or as dedicated global models (GlobalKin [43]), or as open-source codes such as ZDPlasKin [44], PumpKin [45] and the Kinetic Global Model framework (KGMf) [46, 47].

Reaction sets with reaction rate coefficients are incorporated to identify the key reaction pathways and predict discharge behaviors in global models. Reaction rate coefficients can be defined either as look-up tables of a reduced electric field, as analytical functions of electron temperature computed by convolution of the EEDF and cross-sections. Analytical functions are usually created by fitting the experimental results and writing the reaction rate coefficients as Arrhenius formulas that can be scaled using an energy equation. When defined with cross-sections, which can be obtained from experiments (e.g., swarm [48–51] or beam [52] experiments) or from modeling, the reaction rate coefficients are computed by convolution of a cross section and electron energy distribution function (EEDF). The choice of the EEDF has a direct impact on reaction kinetics and the commonly used assumption of a Maxwellian EEDF can lead to a large deviation from the physical distribution [37, 38, 53–55], resulting in incorrectly computed reaction rate coefficients. Conveniently, the EEDF can be defined as a time-independent analytical function of a shape parameter x [56], defined by fitting results from Particle-in-Cell (PIC) [14, 15, 57] or Monte-Carlo [58, 59] simulations. Monte-Carlo simulation tools include BetaBoltz [60], METHIS [61], ELENDIF [62] and MagBoltz [63]. Boltzmann equation (BE) solvers employ approximations of the Boltzmann equation and can be divided into two-term approximation solvers (BOLSIG+ [64], BOLOS [65] and LoKI-B [66]) and multi-term approximation solvers (MultiBolt [67]). The two-term approximation solvers are widely used both for global models and fluid simulations on low-temperature plasmas [53, 68–70].

1.2 The Motivation

1.2.1 Coupling of Self-Consistent Evaluation of the EEDF with a Global Model

Global models are a subgroup of fluid models, where the spatial dependence of input parameters and variables in the system is not considered. Instead, computed average values are beneficial for defining the simulated system’s general properties (behavior). With spatial

dependencies neglected and complicated chemistry included in the modeling, global models' focus is redirected to the collisional term of the continuity equation. The collisional term includes volumetric and surface processes in the system, e.g., creating and destroying particles due to collisions with other species and surface recombination.

The electron energy distribution function (EEDF) has a profound effect on electron reaction rates in the system. In high-power microwave discharges, Kim and Verboncoeur [71] showed that the EEDF shape depends on background gas species and pressure. Nam and Verboncoeur [37, 38, 42] showed the effect of microwave frequency and background gas on the EEDF and consequently on breakdown time in oxygen, xenon, and argon systems in various pressures. Zhao *et al.* [72] benchmarked the validity of the two-term Boltzmann approximation in high-power microwave discharges and showed a large deviation of evaluated EEDF from the Maxwellian EEDF. Comparison of Maxwellian and non-Maxwellian EEDF in He+air system was performed by Sun *et al.* [55] and showed an overestimation of electron probabilities for energies above 22 eV by the Maxwellian EEDF. Including self-consistent evaluation of the EEDF in the global model has obvious advantages but comes with a substantial computational cost [73]. Due to the fast marching in time in global models, the time required to evaluate EEDFs greatly affects the total computation time. Therefore, reducing the EEDF evaluation time or the number of EEDF evaluations is required to successfully integrate the self-consistent evaluation of EEDF into the global model.

1.2.2 Sensitivity Analysis and Global Models

Global models in plasma simulations are known for their fast simulation times, often on the order of minutes, and larger reaction sets can be considered compared to the spatially dependent models. On the other hand, short simulation times gave rise to global models for performing sensitivity analyses which require many model evaluations. Reaction rate coefficients are based on experimental results and have uncertainties attached to their values. Performing the sensitivity analysis with reaction rate coefficients as input parameters can

result in a vast input parameter space. The dimensionality of the input parameter space is equal to the number of reactions in the reaction set and can be several thousand. To fully explore input parameter space, various techniques were developed. Iman and Conover [74] presented a Latin Hypercube Sampling technique with its application to computer models. Morris [75] presented sampling plans to capture elementary and combined effects of uncertainties in computational simulations for models that were too complicated for classical mathematical analysis. Saltelli [76–78] proposed efficient sampling for input parameter space that reduced the number of model evaluations from $O(p^N)$ to $O(pN)$, where N is the number of parameters and p is the number of values for each parameter. Turner [79, 80] investigated complex helium-oxygen chemistry with an assumed Maxwellian EEDF and concluded that these models could be significantly reduced, i.e., 85% reduction in the number of reactions. He classified the 373 reactions of the helium-oxygen chemistry model into three categories according to the level of the reaction’s uncertainty, which was defined based on the source of the rate coefficient (i.e., defined and confirmed by experiments or defined from the theoretical formula).

Sensitivity analysis is a powerful tool for investigating the properties of complex chemistry models. Combining it with global models enables capturing the time-dependent sensitivity of investigated models. The time aspect could be essential in pulse-driven systems, where reaction rates change by many orders of magnitude due to complicated relations between species.

1.2.3 Prioritizing Reaction and Species

Reaction sets can have a large number of species and reactions, Koelman *et al.* [81] presented a chemical model for the splitting of CO_2 with 72 species and 5732 reactions, and Lu & Law [82] presented an iso-octane reaction set with 857 species and 3606 reactions. Reactions and species have different importance in different systems and simulating regimes on the selected quantity of interest (QoI), e.g., species’ density, ignition time, and electron

temperature. With the known importance of reactions and species in a given system, less important reactions can be neglected, reducing the size of the reaction set and consequently reducing the computational complexity of the model.

Prioritizing reactions and species needed for reducing the reaction sets can be reaction-based or species-based. The easiest method to define the importance of individual species is to eliminate species with low densities (Turner [79]) or by finding the short-lived species and generate reaction pathways (Lehmann in 2002 [83] and 2004 [84]). Ding and Lieberman [35, 85] used a global model in connection with PumpKin to define the dominant reaction pathways for He/H₂O discharges. The reaction-based methods are based on many different conditions [86]. A directed relation graph (DRG) method focuses on reactions' contribution to species' density. DRG methods are localized in time, and the reduced reaction set is valid only at a given time. Lu & Law [87] used the DRG method on a detailed ethylene oxidation mechanism with 70 species and 463 reactions. They were able to reduce the reaction set to 33 species and 205 reactions, which can mimic the performance of the detailed mechanism with high fidelity in homogeneous systems of perfectly stirred reactors and diffusive systems of laminar flame propagation. A global skeletal mechanism is generated by combining reduced reaction mechanisms at each time of interest. Lu & Law [88] investigated how reaction sets reduced by the DRG method can recover fast and slow processes, and capturing slow processes required reaction set reduction at multiple time steps.

The importance of individual reactions can be defined according to their relative contribution to species' density based on the reaction rate (in either creation or destruction of the species) or the reaction's contribution to the energy balance of the system, as was demonstrated by Koelman *et al.* [89] using precomputed EEDFs. The reaction rate coefficients are widely accessible through reaction set databases, e.g., LXCat [90] and GRI-MECH 3.0 [91] and the relative contribution to species' density is usually used for defining the importance of reactions. The DRG method can evaluate reaction sets and define the importance of reactions and species on selected species' density. Furthermore, by coupling the DRG method

with a global model to evaluate the importance at different times during the simulation, the resulting reduced reaction set is valid on a broader time interval, which is important in pulse-driven or transient systems. Finally, combining a globally reduced reaction set with self-consistently evaluated EEDF could raise the fidelity of the reduced reaction sets, which could be used in spatially enabled simulations.

1.3 Outline of the Thesis

The thesis is organized as follows. In Chapter 2, the foundations of the Kinetic Global Model framework (KGMf) are introduced, followed by a description of the electron energy distribution function (EEDF) and Boltzmann equation (BE) solvers to self-consistently evaluate the EEDF for use in global models. Benchmarks of the KGMf with coupled BE solver are presented for plasma discharges and high power microwave breakdown. Chapter 3 introduces the sensitivity analysis and selection of input parameters in low-temperature plasma discharges. The global sensitivity analysis (GSA) is applied to pulsed plasma discharges, and the reaction importance using sensitivity indices is presented in argon and H₂-O₂-Ar gas mixtures. Chapter 4 presents methods for prioritizing species and reactions to reduce reaction sets with the goal of reducing computational cost while maintaining fidelity. The directed relation graph (DRG) method is presented in detail, applied to reducing the reaction set in H₂-O₂-Ar plasma discharges with Maxwellian EEDFs and EEDFs from the BE solver. Chapter 5 concludes the material presented in the thesis and presents possible future work.

Important work that does not fit into the main text is presented in the appendices. Appendix A is dedicated to the theoretical models added to the KGMf. Computational additions to the KGMf are presented in Appendix B, which includes testing suit, details about sensitivity analysis runs, and a description of various input and results file formats. Appendix C contains the complete reaction set used in global sensitivity analyses (Chapter 3) and reaction set reduction simulations (Chapter 4).

CHAPTER 2

THE KINETIC GLOBAL MODEL FRAMEWORK (KGMF)

This chapter contains figures and excerpts from a paper by Krek *et al.* [92], published in *Computer Physics Communications*, volume 260, 2021, presenting a benchmark of the KGMf with a coupled Boltzmann equation solver.

2.1 Introduction

The Kinetic Global Model framework (KGMf) [46] is an open-source general-purpose global model simulation code written in Python. The motivation for developing the KGMf was to create a turnkey global model simulation code and as a framework that allows users to modify or add features to fit their modeling needs better. The latter is an important feature that makes the KGMf a powerful educational tool. The ability to add or modify features is important as global models inherently have many approximations. They may vary greatly depending on the physics of the modeled system, e.g., modeling of the sheath at low or high pressure, energy or particle flux losses to the walls, and power absorption. The KGMf can compare reaction data (cross sections and rates coefficients) from different sources in a user-defined model, such as user-defined experimental results and data from other simulation codes and sources (e.g., LXCat [90], ChemKin [93], GRI-Mech [91, 94]). Even more important, the KGMf can compute a system of reaction chains in which the balance of reaction rates and species densities plays an integral role, allowing a better understanding of the connected, holistic system in which limiting steps within reaction chains can be more important than the isolated reaction rates. By converting the ODE system to C code during the simulation initialization and compiling it into a library of callable objects, the simulation time is reduced, and the flexibility of Python is preserved. The compiled object is called from the SciPy integrator during the ODE integration. The object code is created for both

the ODE system matrix and the Jacobian matrix.

The KGMf can be utilized in many ways, from single simulations to sensitivity analyses and reaction set reductions. Since the KGMf is distributed as a source code, internal design and implementation are accessible to the developer. Furthermore, many of the internal methods and objects are accessible to users through an API and can be used when writing a custom Python script for utilizing the KGMf. Every KGMf simulation requires two input files: the simulation file and the reaction file. The simulation file defines the system used in the simulation, background gas pressure, electron and gas temperatures, system volume, absorbed power, electron energy distribution function (EEDF), and initial species densities. The reaction input file defines reactions and their reaction rate coefficients used in the simulation. A reaction rate can be described by a constant, a function of system variables (e.g., electron or gas temperature), or defined with a cross-section. In case the reaction rate is defined with a cross-section, the reaction rate coefficient is computed using equation 2.8.

2.2 The Main Governing Equations

The KGMf is a volume-averaged (0D or global model) simulation code, and all simulation variables and parameters have no spatial information but are time-dependent. The main governing equations in the KGMf are species balance equations, which describe the change of species densities, and the energy balance equation that governs the energy exchange in the system. The species balance equations for each species in the system, k , is defined as:

$$\dot{n}_k = \frac{dn_k}{dt} = \sum_i v_i^k K_i \prod_j n_j + (G - L)_k \quad (2.1)$$

where i is the index of reactions involving species k , K_i is the reaction rate coefficient for reaction i , and j is the product index over all reactant species in reaction i , v_i^k is the integer growth or loss factor for species k from reaction i . The term $(G - L)_k$ is the gain or loss of species k that is independent of reactions, e.g., inflow or outflow of species or surface processes (recombination, secondary emission).

The energy balance equation describes the energy flow between species, species states (e.g., ground, excited, vibrational, rotations, ionized), and the rest of the system (e.g., wall and sheath). The electron energy conservation equation describes the conservation of energy for the electron species and is defined as:

$$\frac{d}{dt} \left[\frac{3}{2} n_e T_e \right] = \frac{P_{\text{abs}}}{V} - n_e \sum_i n_i K_i \Delta E_i + \dot{Q}_e \quad (2.2)$$

where T_e is the electron temperature (or effective temperature for non-Maxwellian EEDF), n_e is the electron density, P_{abs} is the power absorbed by electrons from the electric field, V is the system volume, ΔE_i is the energy threshold of reaction i , and \dot{Q}_e is the energy loss due to electron convection (e.g., sheath losses or losses of electrons to the wall). Each term in Eq. (2.2) can be time-dependent, and the energy transport between different computational species (e.g., ground or excited state atoms, ions, and electrons) is given by reactions composing the reaction set.

Power can be delivered into the system through various mechanisms and is absorbed by a change of species state (e.g., ionization, excitation) or by increasing species energy (electrons). In global models, it is often assumed that all power is deposited to electrons, and power deposition to other species is neglected. Ohm's law can be used in cases when we can assume that power added to the system is deposited into electrons based on their interaction with the applied electric field. The transfer of the energy (power) from the applied electric field to the electrons is possible because the electrons are resisting the movement enforced on them by the electric field. The energy deposited to ions is neglected because heavier ions ($m_i/m_e \approx 1836$) do not respond to the applied electric field due to their large mass and are only affected by the DC component of the electric field. This means that ions are (almost) stationary, and no energy from the applied electric field is deposited to ions. The relation between absorbed power and the electric field is plasma conductivity σ , which is defined by Ohm's law as:

$$\tilde{P}_{\text{abs}} = \sigma E^2 \quad (2.3)$$

where \tilde{P}_{abs} is the absorbed power density, σ is the plasma conductivity, and E is the electric field. The plasma conductivity can be computed with the following equation:

$$\sigma = n_e e \mu_e \quad (2.4)$$

where μ_e is the electron mobility, usually found in tables for steady-state (e.g., Table 2.1 in Ref. [95]) from approximated experimental results, n_e is the electron density, and e is the elementary charge. When electron mobility is not constant, it can be computed from effective collision frequency ν_m as follows:

$$\mu_e = \frac{n_g e}{m_e \nu_m} \quad (2.5)$$

where n_g is the background gas density, m_e is the electron mass, and ν_m is the effective collision frequency computed from the current electron energy distribution function (EEDF) in the system. The EEDF can be defined either with an analytical function or computed by a coupled Boltzmann equation solver. Using Ohm's law for computing deposited power to electrons in simulations involving Boltzmann equation solvers, electron mobility (μ_e) is computed from EEDF and is usually time-dependent.

2.3 Boltzmann Equation and Electron Energy Distribution Function (EEDF)

Boltzmann equation (BE) is one of the most general descriptions of plasma, providing the statistical description of the system's state. For particles driven by an electric field, the Boltzmann equation is [96]:

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla f - \frac{q\vec{E}}{m} \cdot \nabla_v f = \left(\frac{\partial f}{\partial t} \right)_{\text{coll}} \quad (2.6)$$

where $f = f(t, \vec{x}, \vec{v})$ is the spatial-temporal velocity distribution function (VDF) of the species, q is the particle charge, m is the electron mass, $\vec{E} = \vec{E}(t, \vec{x})$ is the electric field, and ∇_v is the convective gradient of f . The $\left(\frac{\partial f}{\partial t} \right)_{\text{coll}}$ term represents the time rate of change in

f due to the collisions and has the same dependencies. The collision term in the BE includes both elastic and inelastic collisions.

Although solving the BE provides a self-consistently computed electron energy distribution function (EEDF) that captures its time evolution, drastic simplifications of the BE are required for computational efficiency. One approach of simplifying the BE is to write the velocity in spherical coordinates with the main velocity oriented in the direction of the applied electric field. With this assumption, any velocity can be approximated using Legendre polynomials with an arbitrary number of terms. For the classical two-term spherical approximation [66, 96] that uses two terms of polynomial approximation and with velocity \vec{v} written in spherical coordinates, the BE becomes:

$$f(t, z, v, \cos \theta) = f_0(t, z, v) + f_1(t, z, v) \cos \theta, \quad (2.7)$$

where t is time, v is the velocity, z is the spatial coordinate along the electric field, θ is the angle between velocity and the z -axis, f_0 is the isotropic, and f_1 is the anisotropic part of the distribution function. The known EEDF and reaction cross-section are used to compute the reaction rate coefficient using the following equation [37]:

$$K_i(t) = \int_0^\infty \sqrt{\frac{2e\varepsilon}{m_e}} \sigma_i(\varepsilon) f(\varepsilon, t) d\varepsilon \quad (2.8)$$

where $\sigma_i(\varepsilon)$ is the energy-dependent cross section for reaction i , ε is the electron energy, $f(\varepsilon, t)$ is the time-dependent EEDF computed from the BE, and e and m are the electron charge and mass, respectively. Cross sections for various species and from different sources (e.g., Biagi, Hayashi, IST-Lisbon, Itikawa, and Morgan) are available via LXCat [90] website.

The validity of the two-term spherical approximation depends on the working gas and the reduced electric field E/N . Phelps and Pitchford [97] investigated anisotropic scattering and its effect on electron transport, which includes the study of the multi-term expansion of Boltzmann equation for higher values of E/N . For electric fields between 10 and 500 Td (1 Td = 10^{-17} V·cm²), the usual two-term approximation does not produce huge errors

and is considered acceptable compared to the six-term approximation. For electric fields between 500 and 1500 Td, the difference between two-term and six-term approximation is about 10 %. For higher values of E/N , the difference in diffusion coefficients becomes larger (> 25 %). The two-term approximation is valid in cases where the ratio between inelastic and elastic collisions is not significant. In the case of a high ratio between inelastic and elastic collisions, including more Legendre terms is necessary (using three up to six terms). They presented a solution of the BE where they preserve an arbitrary number of terms showing the difference to the solution when many terms were removed from the solution, and the difference was between 1 % and 35 %.

The two-term spherical approximation of the BE also assumes a quasi-stationary system. The electric field is assumed to be constant on the time scale of electron collisions, i.e., electron collision frequency is higher than the driving frequency of the electric field. The pressure where the two-term approximation in high-voltage argon pulse discharge is valid can be defined in terms of electron collision frequency as:

$$\nu_e = \bar{v}_e n_g \sigma > \omega \quad (2.9)$$

where σ is the collision cross section, the average electron velocity, \bar{v}_e , is defined as [98]:

$$\bar{v}_e = 6.7 \times 10^5 \sqrt{T_e} \quad (2.10)$$

and the background gas density, n_g , computed with ideal gas law as:

$$n_g = 133 \frac{p}{k_B T_g} \quad (2.11)$$

with p being the background pressure in Torr, T_e is the electron temperature in eV, k_b is the Boltzmann constant, and T_g is the background temperature in K. Plugging Eqs. (2.10) and (2.11) into Eq. (2.9) and solving for pressure, the minimum pressure for the two-term approximation to be valid in argon discharge is defined as:

$$p = \frac{k_b T_g \omega}{133 \sigma 6.7 \cdot 10^5 \sqrt{T_e}} > 0.232 \text{ Torr} \quad (2.12)$$

where the electric field frequency $\omega = 5 \cdot 10^7$ Hz (pulse raise and fall time $\tau = 20$ ns), the average electron energy $T_e = 1$ eV, the gas temperature $T_g = 300$ K, and the collision cross section for argon is $\sigma = 10^{-20}$ m².

2.3.1 Shape Factor x for Analytical Rate Coefficients K_i

Computing the EEDF using the Boltzmann equation presents a computational challenge despite all assumptions and simplifications mentioned in the previous section. An ability to describe the EEDF analytically would significantly reduce the required computational time. The EEDF can be defined as an analytical function of the shape factor x [56]:

$$f(\varepsilon) = \frac{x}{\left(\frac{3}{2}T_{\text{eff}}\right)^{3/2}} \frac{\left[\Gamma\left(\frac{5}{2x}\right)\right]^{3/2}}{\left[\Gamma\left(\frac{3}{2x}\right)\right]^{5/2}} \varepsilon^{1/2} \exp\left\{-\left[\frac{\Gamma\left(\frac{5}{2x}\right)}{\Gamma\left(\frac{3}{2x}\right)} \frac{\varepsilon}{\frac{3}{2}T_{\text{eff}}}\right]^x\right\} \quad (2.13)$$

where x is the shape factor, Γ is the Gamma function ($\Gamma(z) = \int_0^\infty y^{z-1} e^{-y} dy$), and ε is the electron energy (in eV). The shape factor x is not limited to certain values, can be a real or integer number and can be used to describe common distribution functions: $x = 1$ for Maxwellian and $x = 2$ for Druyvesteyn EEDF (Fig. 2.1).

Equation (2.13) provides a simplified way to capture the impact of EEDFs on plasma discharge characteristics and to predict the truncated EEDFs through a unified equation with only two parameters (a shape factor x and an effective electron energy T_{eff}). The approximation has limited use in systems showing bi-Maxwellian or bump-on-tail (beam-plasma systems) EEDFs, as some EEDF shapes cannot be fully captured with this approximation. The validity of the EEDF approximation by Eq. (2.13) strongly depends on the simulated system. In Ref. [38], the value of the shape factor x showed a weak dependence on pressure and strong dependence on species (working gas). In the case of high-power microwave breakdown, the dependence on the driving frequency of the electric field can be observed. The two critical conditions for validity of the shape factor approximation are:

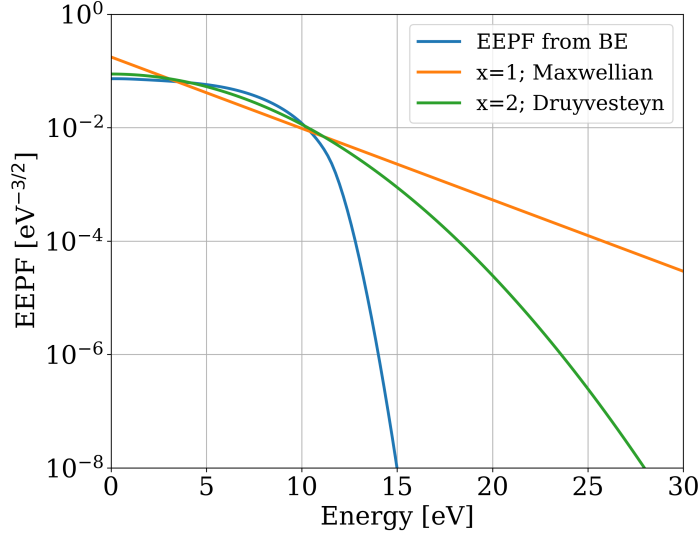


Figure 2.1 Plots of different EEDF shapes: from simulations (EEPF from BE) and from Eq. (2.13) with different shape factors x .

1. the EEDF is well represented by a single shape factor x , e.g., single temperature EEDFs,
2. the EEDF must remain relatively constant in time unless a time-dependent shape factor $x = x(t)$ is used.

When these conditions are violated, a more complicated representation may be needed, up to the use of a dynamic Boltzmann solver. The current implementation in the KGMf does not support the time-dependent shape factor $x(t)$ and the EEDFs defined by shape factor x cannot be time dependant.

2.3.2 Boltzmann Equation Solvers

The solution of the Boltzmann equation (BE) for electrons is the electron energy distribution function (EEDF), which is an important input parameter for global models. The EEDF is used to compute reaction rate coefficients from cross-sections using Eq. (2.8). Many different BE solvers are accessible under open-source or free-software licenses. They use either

a spherical approximation of the Boltzmann equation or use the Monte-Carlo method to track electrons to compute the EEDF for non-magnetized low-temperature plasmas. The BE solvers using spherical approximation can use either two-term approximation (e.g., BOLSIG+[96], BOLOS [65], and LoKI-B [66]), or multi-term approximation (e.g., MultiBolt [67]). Solvers that use the Monte-Carlo method include BetaBoltz [60], METHIS [61], and MagBoltz [63]. When solving the Boltzmann equation assuming local equilibrium, the main input parameters for most BE solvers are the reduced electric field (E/N), reaction cross-sections, and species density ratios. Cross-sections are required in tabular or equation form for every species and reaction (e.g., moment transfer, ionization, excitation, and de-excitation) in the system. The LXCat [90] website provides access to cross-section data for a large number of species and from many different sources.

2.4 Coupling the KGf with Boltzmann Equation Solvers

The Boltzmann equation (BE) solver is coupled to the KGf via `EEDFsolver` class that wraps the actual implementation of the BE solver and acts as a layer between the KGf and BE solver, as shown on Fig. 2.2. Adding a wrapper around BE solvers enables the KGf to compare coupled BE solvers (e.g., BOLOS and MultiBolt) and enables future upgrades, such as coupling with LoKI-B or BOLSIG+. This flexibility enables users to select the BE solver that best fits their needs in terms of accuracy and computational complexity (two- or multi- terms). The coupled BE solver is used during the simulation only when selected in the simulation input file.

The communication between the KGf and the BE solver is done at the simulation initialization and every time step during the simulation. At the simulation initialization (*KGf init* on Fig. 2.2) the `EEDFsolver` class and the underlying solver are initialized. The initialization involves extracting electron impact reactions (i.e., reactions with electrons as reactants) defined with cross-sections, loading required cross-sections, and creating a list

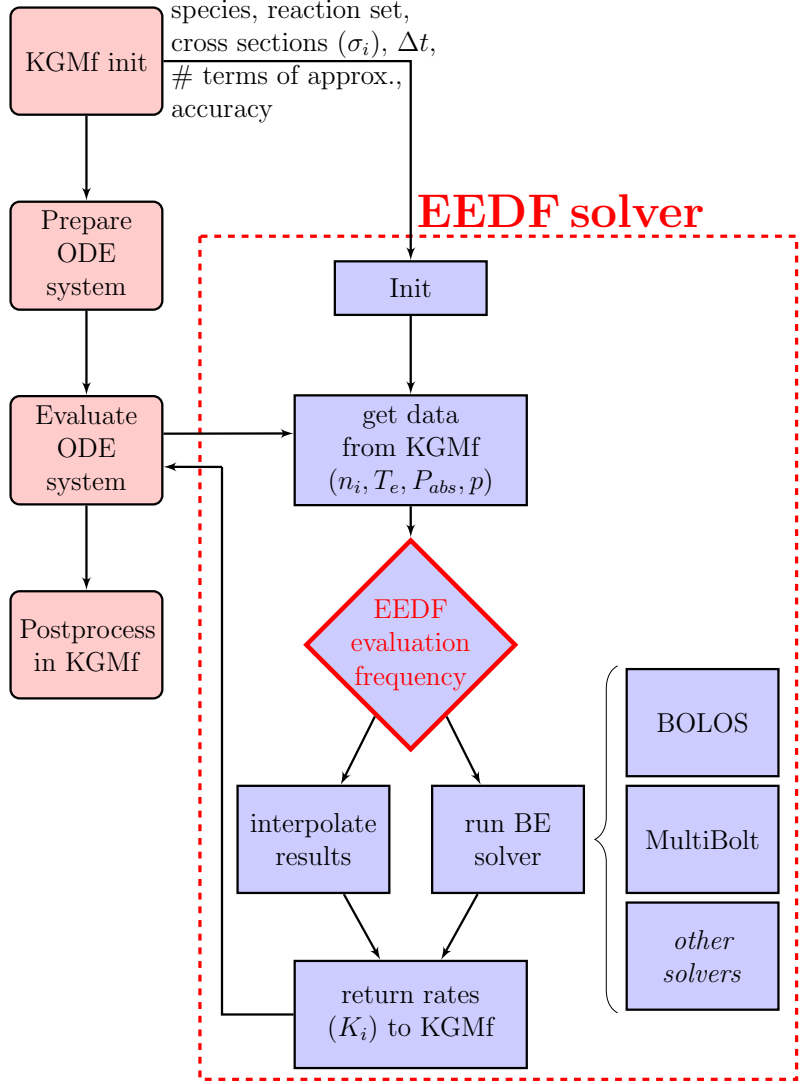


Figure 2.2 Flowchart of the KGMf and BE solver coupling via `EEDFsolver` class. The `EEDFsolver` class is initialized according to selected simulation parameters. In the time advancement of the ODE system, parameters are set in the `EEDFsolver` class, and the decision is made whether to call the selected BE solver to evaluate the EEDF or use the previously computed EEDF to compute reaction rate coefficients.

of non-electron species in those reactions. The species list in the BE solver is a subset of the species list in the KGMf, as it only contains non-electron species acting as reactants in electron impacted reactions. Depending on the selected BE solver, the number of terms for the approximation is defined. After the initialization, the BE solver is run for the first time to compute the initial EEDF.

In every time step in the simulation, the `EEDFsolver` class is called from within the ODE integrator (*Evaluate ODE system* on Fig. 2.2). According to the selected EEDF evaluation frequency, the decision is made in the `EEDFsolver` class to either compute rate coefficients using the existing EEDF or to compute the new EEDF, and new rate coefficients are computed using the new EEDF. The EEDF evaluation frequency parameters are defined in the input file. In either case, the reaction rate coefficients are returned to the ODE evaluation routine. If the EEDF needs to be evaluated, the species density ratios, reduced electric field, and the EEDF energy grid are sent to the BE solver. The BE solver returns the new EEDF on a given energy grid. When the value of the EEDF at $0.8\varepsilon_{\max}$ is below 10^{-13} , where ε_{\max} is the upper limit of the currently defined energy grid, the energy grid is adjusted, and the EEDF is re-evaluated using the new energy grid. The grid can be defined either as a linear grid in the whole energy range (with a fixed number of grid points) or as a multi-interval linear grid, where the whole energy range is divided into linear grid intervals (with a fixed number of grid points in each interval). In the former case, the energy mapping of the EEDF will increase the accuracy of the EEDF evaluation at lower energies by reducing the grid step size. In the latter case, the mapping will reduce the computational load when computing reaction rate coefficients by reducing the total number of grid points.

The implementation of the `EEDFsolver` class was verified for EEDF calculation and its evaluation frequency. For given values of species density ratios and reduced electric field, the EEDF was evaluated and compared to the EEDF evaluated by BOLSIG+ for the same input parameters. The second check was the EEDF evaluation frequency, where conditions for invoking the EEDF evaluation were checked to ensure the EEDF was evaluated at the correct relative change of selected system parameter, e.g., reduced electric field or electron density. Evaluation frequency conditions, the parameter used, and its threshold value is defined in the simulation input file.

2.4.1 EEDF Evaluation Frequency

Evaluating the EEDF with coupled BE solver increases the fidelity of results and increases the simulation's total run time. There are times when the EEDF does not change significantly during the individual time steps, and the already computed EEDF from the previous time step could be used in the current time step. This indicates that the EEDF could be self-consistently evaluated at different frequencies and not necessary every time step. The frequency for EEDF evaluation depends on a simulated system. The apparent parameters on which to base the evaluation frequency would be input parameters to the BE solvers or result from BE solvers, such as reduced electric field (E/N), electron density (n_e), electron temperature (T_e), and gas mixture ratio. According to the available documentation, BOLSIG+ [96] uses caching mechanism, based on reduced electric field E/N , to reduce the number of necessary evaluations. BOLSIG+ uses a geometric grid of reduced electron field E/N values to create a history of evaluated EEDFs and to decide whether to evaluate the EEDF for a given E/N or reuse the previously evaluated EEDF.

In the KGMf, evaluation frequency is implemented based on the number of steps (evaluate every given number of steps) or on relative changes of plasma parameters in the system: reduced electric field (E/N), electron density (n_e), or electron temperature (T_e). The parameter and its threshold value for defining the EEDF evaluation frequency are user-defined in the KGMf input files. When EEDF evaluation is requested, the value of the selected parameter is compared to the value at the last EEDF evaluation. If the relative difference between two values is bigger than the threshold value, the EEDF is evaluated; otherwise, the previously evaluated EEDF is used. For example, the evaluation condition for electron temperature is defined as:

$$\frac{|T_e - T_{e,eval}|}{T_{e,eval}} > \Delta T_{e,thr} \quad (2.14)$$

where T_e is the electron temperature at current evaluation, $T_{e,eval}$ is the electron temperature at the last evaluation and $\Delta T_{e,thr}$ is the electron temperature threshold given as an

input parameter (e.g., 0.1 for 10%, 0.4 for 40%). When the condition is satisfied, the EEDF is evaluated using a coupled BE solver, and the updated EEDF is used until the next EEDF evaluation. Currently, only one threshold parameter can be specified as an evaluation frequency condition. The effect of EEDF evaluation frequency on results' accuracy and total simulation run time in benchmark cases of the KGMf and ZDPlasKin, is presented in Section 2.5.2.

2.4.2 Coupled BE Solvers

The KGMf is currently coupled with two BE solvers: BOLOS (two-term BE solver) and MultiBolt (multi-term BE solver). BOLOS was benchmarked against ZDPlasKin and is more extensively used, tested, and verified. Users can select the BE solver for their simulations with the setting in the simulation input file.

BOLOS [65] is an open-source and freely accessible implementation of a two-term BE solver that is similar to BOLSIG+. BOLOS is written in Python and can be included as a module to any Python-based script. Before coupling BOLOS with the KGMf, some modifications to BOLOS were necessary, mainly in places where BOLOS interacts with the KGMf: transferring parameters to BOLOS (e.g., cross-sections and species densities) and gathering the results from BOLOS (e.g., EEDF, electron temperature, and electron mobility).

MultiBolt [67] is a multi-term spherical approximation BE solver that is supporting from two and up to 10-term Boltzmann equation approximations. It is implemented in MatLab and is freely distributed as a source code. The main parameters for MultiBolt are the reduced electric field, reaction cross-sections, and species density ratios. Coupling of MultiBolt with the KGMf required significantly more modifications than BOLOS since it involved preparing MultiBolt to run inside Octave (MatLab-like environment in Linux). Once MultiBolt successfully ran in Octave, the KGMf was able to run it through the **oct2py** module. Modifications to the original code were made to accommodate the different setup of input parameters from the KGMf and returning computed results to the KGMf.

2.5 Benchmark of the KGMf with Coupled Boltzmann Equation Solver

The KGMf coupled with BOLOS was benchmarked with ZDPlasKin with coupled BOLSIG+ to verify the implementation and coupling of the BE solver. ZDPlasKin requires the user to define the main loop, initial values of species densities, and the time evolution of independent parameters. The simulation reaction set from the input file is converted into FORTRAN source code and compiled with the rest of the source files into an executable file, which results in highly optimized and fast simulation code. ZDPlasKin does not include the electron energy balance equation, and the effective electron temperature is computed using the EEDF from BOLSIG+ with the following equation: $T_e = \frac{2}{3} \int_0^\infty \varepsilon f(\varepsilon) d\varepsilon$. To be able to compare the results from both codes, the capability to compute T_e from a coupled BE solver was added to the KGMf.

The benchmarking was done for high pressure low-temperature argon plasma discharge cases without considering the sheath and neglecting all species or energy losses to the wall and particle-field energy exchange in the sheath. These assumptions have a negligible effect for plasmas in which the bulk region is much larger than the sheath region but fail when the size, and particularly the energetics of the sheath/wall regions, become comparable to that of the bulk region. The reaction set used for benchmark included ionization, single-state excitation, recombination, and step-wise ionization and is presented in Table 2.1. Reaction rate coefficients for reactions R1-R3 are defined as cross-sections and as Arrhenius form as functions of electron temperature T_e for reactions R4-R9. The species considered in the reaction set include ground state atoms (Ar), electrons (e), excited state atoms (Ar*), atomic ions (Ar⁺), and molecular ions (Ar₂⁺). Initial densities for electrons (n_e) and atomic ions (n_{Ar^+}) were 10^7 cm^{-3} , densities for molecular ions ($n_{\text{Ar}_2^+}$) and excited atoms (n_{Ar^*}) were 10^2 cm^{-3} .

Table 2.1 The reaction set for the benchmark case. The reaction set considers the following species: ground state atoms (Ar), electrons (e), excited state atoms (Ar*), atomic ions (Ar⁺), and molecular ions (Ar₂⁺). *Notes:* T_e in eV and T_{gas} in K.

Reaction	Rate coefficient, unit	ΔE_i [eV]	Refs.
R1: Ar + e \rightarrow Ar ⁺ + 2e	cross section	15.76	[99]
R2: Ar + e \rightarrow Ar* + e	cross section	11.50	[99]
R3: Ar* + e \rightarrow Ar ⁺ + 2e	cross section	4.30	[99]
R4: Ar ₂ ⁺ + e \rightarrow Ar* + Ar	$8.5 \times 10^{-7} (11600 T_e/300.0)^{-0.67}$, cm ³ /s	$1.5 T_e$	[44]
R5: Ar ₂ ⁺ + Ar \rightarrow Ar ⁺ + 2Ar	$6.06 \times 10^{-6}/T_{\text{gas}} \exp(-15130.0/T_{\text{gas}})$, cm ³ /s		[44]
R6: Ar* + Ar* \rightarrow Ar ₂ ⁺ + e	6×10^{-10} , cm ³ /s	$-1.5 T_e$	[44]
R7: Ar ⁺ + 2e \rightarrow Ar + e	$8.75 \times 10^{-27} T_e^{-4.5}$, cm ⁶ /s	$1.5 T_e$	[44]
R8: Ar* + 2Ar \rightarrow Ar + 2Ar	1.4×10^{-32} , cm ⁶ /s		[44]
R9: Ar ⁺ + 2Ar \rightarrow Ar ₂ ⁺ + Ar	$2.25 \times 10^{-31} (T_{\text{gas}}/300.0)^{-0.4}$, cm ⁶ /s		[44]

For this benchmark, constant absorbed electron power density ($\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$), system volume ($V = 1 \text{ cm}^3$), and ion temperature (T_i) were assumed. There was no inflow or outflow of species (closed system), and all losses to the wall were neglected. The absorbed power was deposited only to electrons via Ohmic heating ($\tilde{P}_{\text{abs}} = e \mu_e n_e E^2$). With high collision frequency in DC discharges at high pressure for weakly ionized plasmas, Ohmic heating is the dominant heating mechanism [95, 98]. Coupled BE solvers (BOLOS and BOLSIG+) were used to compute electron mobility (μ_e) and to evaluate the dynamic EEDF self-consistently.

The reduced electric field E/N is the main input parameter for many BE solvers. With assumed Ohmic heating and constant absorbed power density, the E/N is computed as:

$$\left(\frac{E}{N}\right) = \frac{1}{N} \left(\frac{\tilde{P}_{\text{abs}}}{n_e e \mu_e}\right)^{1/2} \quad (2.15)$$

where Ohmic heating is defined as $\tilde{P}_{\text{abs}} = e \mu_e n_e E^2$, N is the background gas density, $\tilde{P}_{\text{abs}} = P_{\text{abs}}/V$ is the absorbed power density, n_e is the electron density, e is the elementary charge, and μ_e is the electron mobility.

The EEDF is computed by coupled BE solvers self-consistently, BOLOS in the KGMf, and BOLSIG+ in ZDPlasKin. Both BE solvers use the two-term approximation for solving the Boltzmann equation, which is not valid for every value of the reduced electric field and

could lead to errors in the computed EEDF. The KGMf enables the use of different EEDF evaluation frequencies of coupled BE solvers to reduce the computational cost and reduce the total run time, possibly impacting computed plasma parameters. The EEDF evaluation frequency is defined based on changes in system parameters. Generally, a larger change in system parameters between EEDF evaluations would reduce the fidelity of the EEDF during discharge evolution.

2.5.1 Temporal Evolution of Plasma Parameters

The temporal evolution of the reduced electric field E/N and electron temperature are benchmarked first since they are the key to most of the other discharge parameters. Figure 2.3 shows the reduced electric field E/N and the effective electron temperature T_e as a function of time, from an initial towards a steady state. Due to the constant absorbed power density and Ohmic heating, the reduced electric field decreases with time as the electron density (n_e) increases until it reaches the steady-state. The electron temperature is decreasing with time from the initial value of 10 eV to 3.5 eV in the steady-state. The main drop of electron temperatures is between 10^{-12} and 10^{-9} s that corresponds to a large change of the

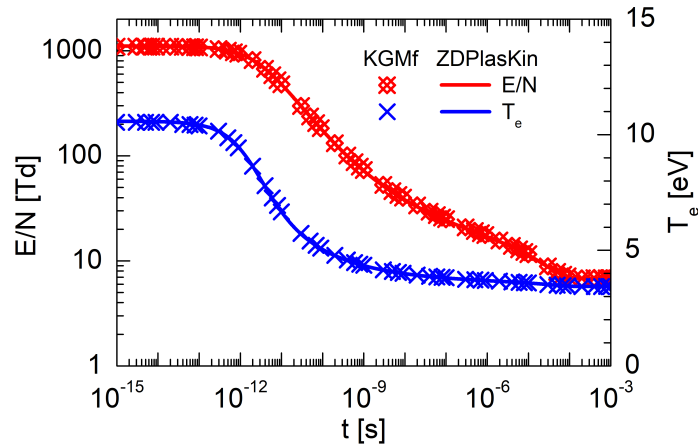


Figure 2.3 Temporal evolution of the reduced electric field E/N and the effective electron temperature T_e . Pressure $p = 760$ Torr in argon and absorbed power density $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$.

EEDF shape during the same time interval. The steady-state time is case dependent, and in our case, the system reached the steady-state at $t = 10^{-4}$ s, from where the main plasma parameter values (e.g., E/N , T_e , and species densities) stay constant with time. The relative differences at the steady-state for electron temperature (T_e) and reduced electric field (E/N) were 0.06% and 0.4%, respectively.

Figure 2.4 shows the temporal evolution of EEDFs, where the first two series of curves present EEDFs computed by BE solvers, and the third series of curves is from Maxwellian EEDFs. The EEDFs from the KGMf and ZDPlasKin show good temporal agreement, confirming the effective electron temperature agreement. The drastic changes in the EEDF shapes between 10^{-12} and 10^{-9} s correspond to considerable changes in the respective electron temperatures, as can be observed in Fig. 2.3. The Maxwellian EEDFs are plotted using the effective electron temperature from the EEDF at each plotted time. The shape of the Maxwellian EEDF deviates drastically from the EEDF computed from BE solvers and shows

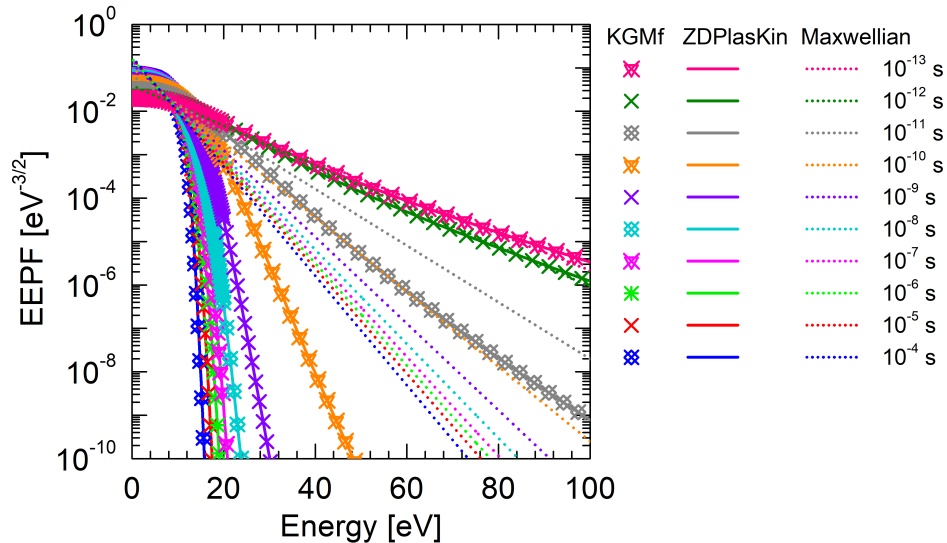


Figure 2.4 Temporal evolution of the EEDFs from BOLOS (KGMf), BOLSIG+ (ZDPlasKin) and analytical Maxwellian EEDF (KGMf) for the same effective electron temperature (T_e) and at the same times. Maxwellian EEDFs (dotted lines) show higher probability of electrons with energies above ~ 20 eV compared to EEDFs from BOLOS and BOLSIG+ at each presented time. The pressure $p = 760$ Torr and the absorbed power density $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$.

a considerable error in energy distribution for times after 10^{-11} s and energies above 20 eV. The results also suggest a much different treatment of inelastic collisions, which affects the EEDF tail and explains why the Maxwellian EEDF overpredicts the electron impact reactions and the steady-state electron densities.

Figure 2.5 shows temporal evolutions of densities of electrons, ions, and excited species. With a constant absorbed power deposited in the system, the species densities change until the generation and the destruction of species balance out at a steady state. The species density evolution shows a good agreement in the transient and at the steady-state. The relative differences in species densities at steady state (from $t=10^{-4}$ to 10^{-3} s) are shown in Table 2.2. The ionization degree at steady state is $\chi = 1.05 \times 10^{-8}$. Figure 2.5 shows the conversion of the dominant ions, changing from the atomic ions at the earlier stage to the molecular ions towards the steady-state, which is consistent with the results in Ref. [41].

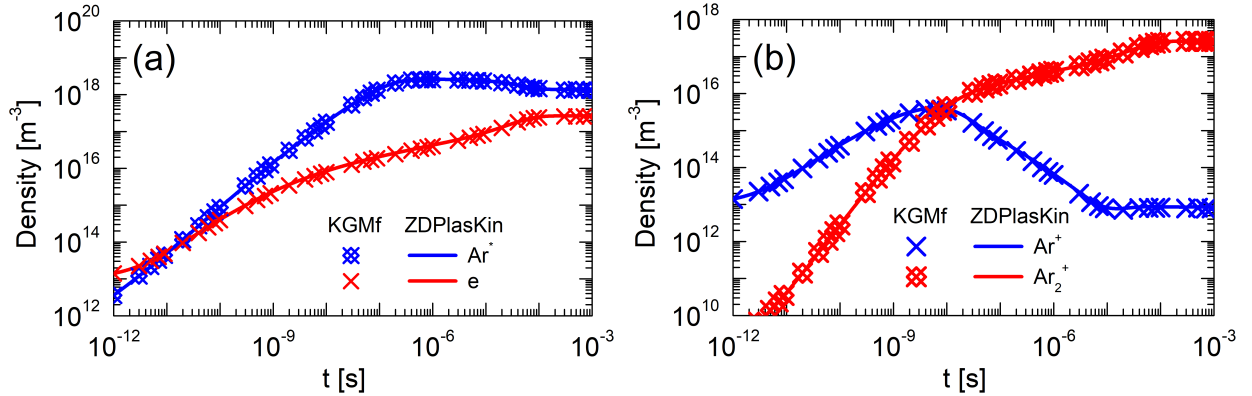


Figure 2.5 Time dependence of species densities: (a) electron and excited species, (b) the atomic and molecular ion densities. Steady-state background gas density $n_g = 2.44 \times 10^{19} \text{ cm}^{-3}$ and the ionization degree $\chi = 1.05 \times 10^{-8}$. The pressure is $p = 760 \text{ Torr}$ and the absorbed power density is $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$.

The temporal evolution of reaction rates from the KGMf and ZDPlasKin are compared in Fig. 2.6 and show good agreement between codes in the whole time interval. The general trend of reaction rates shows they stabilize with time and eventually reach a steady state. Figure 2.6(a) shows reaction rates for rate coefficients defined with cross-sections and

Table 2.2 The impact of different changes of the reduced electric field E/N (used to define the EEDF evaluation frequency in the KGmf) on total computational times and steady-state densities. Total running times for the KGmf (t_K) and ZDPlasKin (t_Z) are given in seconds. The percent difference between the KGmf and ZDPlasKin is given for the following species densities: electrons (e), excited state atoms (Ar^*), atomic ions (Ar^+), and molecular ions (Ar_2^+).

$\Delta E/N$ [%]	t_K [s]	t_Z [s]	t_K/t_Z [-]	e [%]	Ar_2^+ [%]	Ar^+ [%]	Ar^* [%]
0.5	73.52	9.18	8.01	12.06	12.06	22.16	11.84
1	39.40	9.18	4.29	11.54	11.54	21.03	11.31
2	11.68	9.18	1.27	13.98	13.98	25.94	13.27

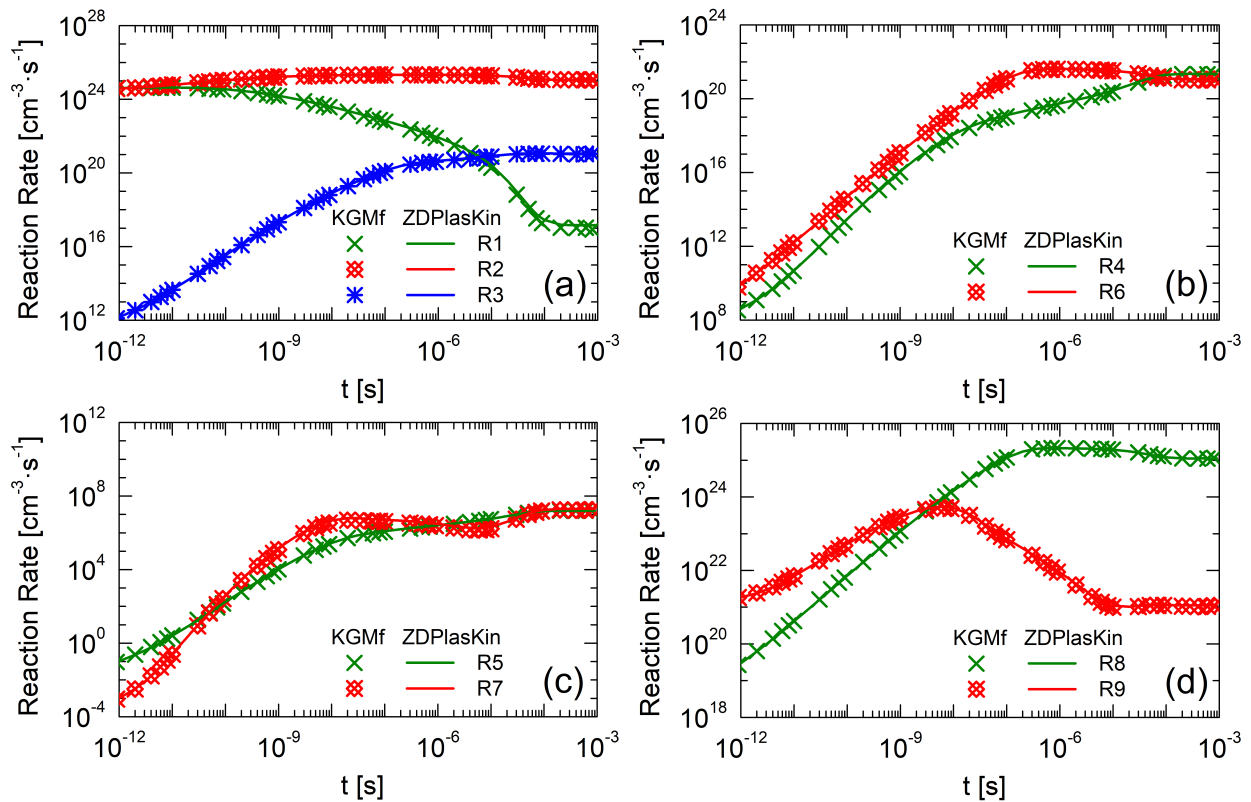


Figure 2.6 Time evolution of reaction rates for (a) reactions defined by cross sections and (b)-(d) reactions defined in Arrhenius form. Reaction labels are from Table 2.1. The pressure is $p = 760$ Torr and the absorbed power density is $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$.

computed using the EEDF from BE solvers. The ionization rate R1 decreases with time because the EEDF shape is becoming steeper, with fewer high energy electrons (losing the high energy tail, see Fig. 2.4), although the electron density increases. The excitation rate, defined by reaction R2, is the main pathway for the creation of excited state species (Ar^*) and shows small changes in time until it is balanced out by reaction rate R8 at a steady state. The stepwise ionization from a single excited state (all excited argon levels combined into a single aggregate excited level), defined by reaction R3, increases with time mainly due to the increasing excited species density and lower threshold energy compared to ionization and excitation. Figures 2.6(b)-(d) show reaction rates for analytically defined reaction rate coefficients. From Table 2.1, we know the reaction rate R4 (destruction of Ar_2^+) and rates R6 and R9 (creation of the Ar_2^+) are the main reactions driving the density of Ar_2^+ . The reaction rate R9 is a conversion of atomic (Ar^+) to molecular (Ar_2^+) ion. The background gas density is constant, and the change of the reaction rate R9 depends only on Ar^+ density. Before reaching the steady-state, the reaction rate R9 is much higher than that of reactions R4 and R6 combined, which indicates that three-body collisions are the key pathway of converting Ar^+ to Ar_2^+ . In Fig. 2.6(c), it can be observed that the reaction R5 (conversion of Ar_2^+ to Ar^+) is less important as its rate is many orders of magnitude smaller compared to reaction rates of reactions R4 and R9. The same trend can be observed by comparing the trend of Ar^+ density and reaction rate R9 as they both first increase until about 10^{-8} s and then start decreasing until reaching the steady-state. This is also consistent with the density evolution of Ar^+ and Ar_2^+ shown in Fig. 2.5(b). Figure 2.6(d) shows the reaction rate R8 (destruction of Ar^*) is increasing in time until it balances the reaction rate R2 (creation of Ar^*) at a steady state.

2.5.2 Computational Implication of Including a BE Solver

Capturing the temporal evolution of the EEDF using a BE solver comes at the cost of longer run times. The EEDF evaluation frequency can be defined in terms of changes of plasma

parameters, such as reduced electric field and species densities, to lower the computational intensity of the coupled BE solver. The EEDF evaluation frequency has an impact on simulation run time and error in computed parameters.

For the presented benchmark case with pressure $p = 760$ Torr and absorbed power density $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$, the EEDFs in the KGMf were evaluated at 1% change of the reduced electric field E/N . This resulted in 847 EEDF evaluations in 10899 time steps, evaluating the EEDF, on average, in 7.7% of time steps. The total run times were 39.4 s for the KGMf and 9.18 s for ZDPlasKin (with EEDF caching enabled in BOLSIG+), making the KGMf 4.3 times slower than ZDPlasKin. ZDPlasKin does not enable the setting of EEDF evaluation frequency but instead relies on the EEDF caching mechanism in BOLSIG+ to increase the computational efficiency (i.e., reducing the overall computational time). The run time of the benchmark case in ZDPlasKin with EEDF caching disabled in BOLSIG+ was 116.8 s, which makes ZDPlasKin 2.9 times slower compared to the KGMf.

More detailed profiling of both codes reveals that the KGMf spends 65% of the total run time in BOLOS, while ZDPlasKin spends 90% of the total run time in BOLSIG+. While both codes spend most of the computation time in BE solvers, the difference in overhead is significant, particularly at high BE evolution frequency. Large overhead in the KGMf could be attributed to the fact that the KGMf has a more feature-rich API than ZDPlasKin's minimalistic approach in which the user controls the code. Part of the overhead is also due to different underlying languages, Python for the KGMf and FORTRAN for ZDPlasKin, as each offers different optimization capabilities. Reducing the overhead and optimizing the code will be one of the future extensions of the KGMf.

2.5.3 Steady-state Densities

Time evolution of species densities for given absorbed power density and atmospheric pressure shows good agreement between codes. The steady-state occurred around time $t = 10^{-4}$ s when densities became constant. The steady-state densities were compared for two series of

runs, the first run with a fixed absorbed power density $\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$ and pressure p in a range from 100 to 1000 Torr, and the second run with a fixed pressure $p = 760 \text{ Torr}$ and absorbed power density \tilde{P}_{abs} in range from 50 to 1000 $\text{W}\cdot\text{cm}^{-3}$.

Figure 2.7 shows steady-state densities for a fixed power and varying pressure. The reduced background gas density at lower pressure reduces the collision frequency, increasing the energy gain between collisions, resulting in more high-energy electrons. High-energy electrons increase the possibility of ionization and cause higher steady-state densities of ions and electrons. The pressure 100 Torr still presents a high-pressure regime as the molecular ions are the dominant ions. However, the tendency of densities suggests that below some pressure, the atomic ions will be the dominant ions [41].

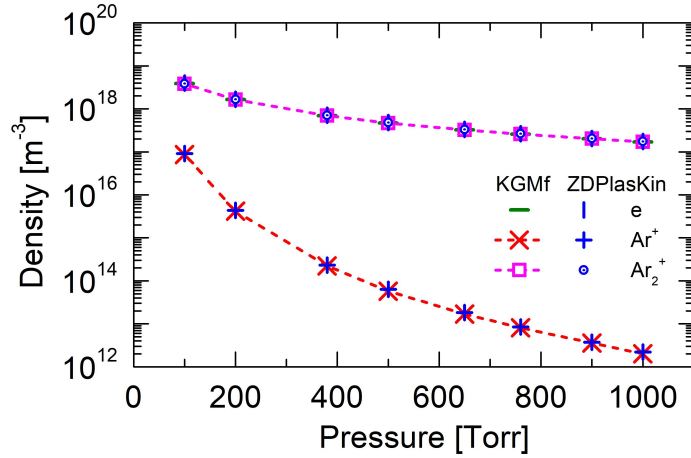


Figure 2.7 The steady-state results for fixed deposited power density ($\tilde{P}_{\text{abs}} = 50 \text{ W}\cdot\text{cm}^{-3}$) and a range of gas pressures ($p = 50 - 1000 \text{ Torr}$).

Figure 2.8 shows the dependence of the steady-state densities on the absorbed power density at 760 Torr. The steady-state densities of all species increase with increasing absorbed power density due to the ionization rate increasing from Eq. (2.8). The molecular ions are the dominant ions for this high-pressure regime, and the electron densities are almost the same as molecular ion densities due to observed quasi-neutrality. The Ar_2^+ densities are $10^4 - 10^5$ times higher than Ar^+ densities, which is mainly caused by the three-body collisions (R9 reaction) converting atomic ions to molecular ions.

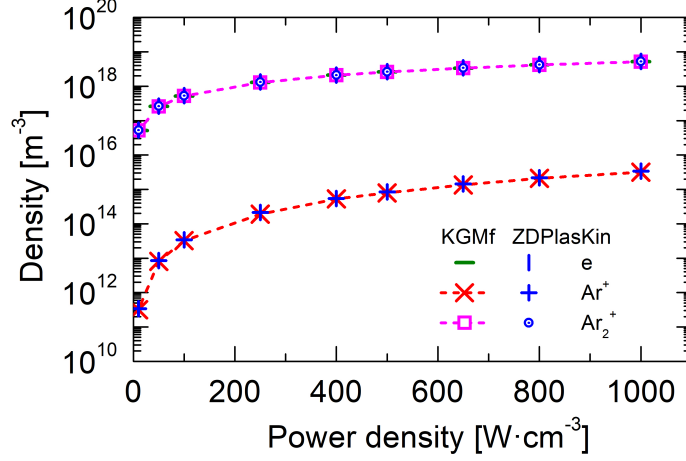


Figure 2.8 The steady-state results for fixed pressure ($p = 760$ Torr) and a range of deposited power densities ($\tilde{P}_{\text{abs}} = 10 - 1000 \text{ W}\cdot\text{cm}^{-3}$).

The KGMf and ZDPlasKin, both coupled with BE solvers, show good agreement in the temporal evolution of plasma parameters, such as the species densities, reduced electric field, and effective electron temperature, in high pressure low-temperature argon plasmas. The agreement of the densities at a steady-state with a fixed power density and a fixed pressure is additionally confirmed. The temporal evolution of EEDFs, computed from BOLOS and BOLSIG+, also agrees well with each other, showing highly truncated non-Maxwellian EEDFs towards the steady-state.

2.5.4 High Power Microwave Breakdown

Gas breakdown characteristics are of great interest for predicting plasma formation under various conditions, which are primarily affected by the surface emission and bulk ionization [100–102]. In high power microwave (HPM) systems, the electron impact collisions with background gas are the key ionization mechanism during the breakdown process, which can be predicted based on the global model [42]. The Maxwellian EEDF, usually assumed in global models, leads to shorter breakdown times at high pressure due to the high energy tail of the EEDF. The KGMf with coupled BE solver can evaluate the EEDF self-consistently and predict breakdown times with higher fidelity; capturing the truncated EEDF yields more

accurate electron impact rate coefficients and electron densities, which is significant for modeling diagnostics. The breakdown times in the KGMf and ZDPlasKin were defined, following the literature [37, 38, 42, 103], as the time required for the electron density to increase by 10^8 times the initial electron density, which was 10^7 cm^{-3} in this case. The reaction set from Ref. [37] (momentum transfer, ionization, and excitation) was used to benchmark the KGMf and ZDPlasKin with results from PIC simulation. BE solvers in both codes computed the reaction rate coefficients for all three reactions. It should be noted that it is not necessary to consider molecular ions since, at the early stage of the breakdown, the atomic ions are still dominant, and the three-body conversion of Ar^+ to Ar_2^+ would occur after the breakdown.

With microwave field as an input parameter, the effective microwave field was computed with [42]:

$$E_{\text{eff}} = \frac{E_0}{\sqrt{2}} \frac{\nu_m}{\sqrt{\nu_m^2 + \omega^2}} \quad (2.16)$$

where E_0 is the amplitude and ω is the angular frequency of microwave field, and ν_m is the momentum transfer collision frequency computed by BE solvers (BOLOS and BOLSIG+).

The reduced electric field needed by BE solvers was computed assuming Ohmic heating of electrons and the absorbed power computed using the following equation [42]:

$$P_{\text{abs}} = \int_0^\infty \frac{e^2 n_e}{m_e \nu_m} E_{\text{eff}}^2 f(\varepsilon) d\varepsilon \quad (2.17)$$

where e is the electron charge, n_e is the electron density, m_e is the electron mass, $f(\varepsilon)$ is the EEDF from coupled BE solvers, and ε is the electron energy.

In Ref. [37], it was suggested that the breakdown time dependence on pressure could be divided into two pressure ranges: below 10 Torr (low pressure) and above 10 Torr (high pressure). In the low-pressure range, where the reduced electric field is high (above 7000 Td and increasing with decreasing pressure), the EEDF has a high energy tail similar to the Maxwellian distribution. In the high-pressure range, where E/N decreases with increasing pressure, the EEDF gets truncated and deviates from a Maxwellian distribution. This behavior can be observed in Fig. 2.9, where the breakdown times predicted by Maxwellian

EEDF (KGMf $x = 1$) and self-consistent EEDF (KGMf with BOLOS) start deviating at pressures above 100 Torr.

The KGMf with the prescribed shape of EEDF ($x = 6.5$) correctly captured the proper tendency of the breakdown times. The results in Fig. 2.9 suggest that BE solvers can be cost-effective only for pressures above 10 Torr, as below 10 Torr, the breakdown times are the same as with assumed Maxwellian EEDF. Both PIC and global model (the KGMf and ZDPlasKin with coupled BE solver) results in Fig. 2.9 show the same tendency of breakdown time at different pressures. The difference in predicted breakdown times could be attributed to spatial dependence of plasma parameters in PIC simulations, mainly electron density used to define the breakdown time. EEDFs used in global models neglect non-local effects, which are inherently included in the PIC simulations, e.g., surface processes and processes in the sheath. At pressures above 20 Torr, the electron density in the PIC simulation is not spatially uniform, and locally higher electron density could cause shorter breakdown times than the breakdown time based on a globally averaged electron density used in the KGMf and ZDPlasKin.

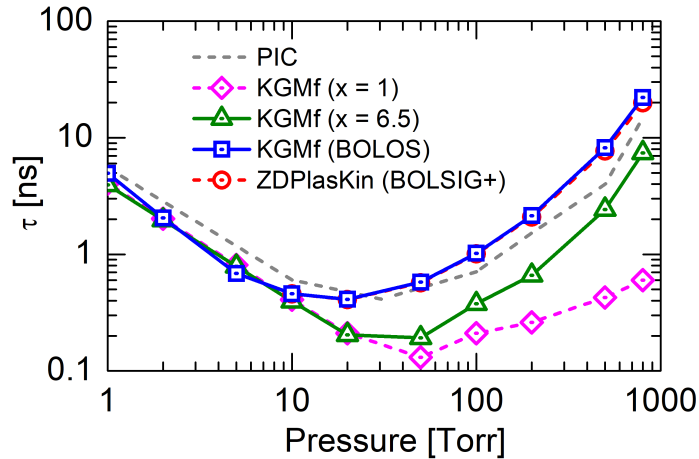


Figure 2.9 Comparison of breakdown times in high power microwave argon discharges for PIC simulation and global models with EEDF defined analytically ($x = 1$ and $x = 6.5$) and BE solvers (BOLOS and BOLSIG+). Analytically defined EEDFs were constant in time, while EEDFs evaluated by the BE solvers were time-dependent. The electric field amplitude $E_0 = 2.82$ MV/m and the angular frequency $\omega = 2.85$ GHz. The PIC results were taken from Ref. [37].

CHAPTER 3

GLOBAL SENSITIVITY ANALYSIS

3.1 Uncertainty Quantification and Sensitivity Analysis

The objective of uncertainty quantification is to quantify the effect of model uncertainties on model outputs. Model uncertainties arise from different sources: model errors, input uncertainties, and numerical implementation of the model [104]. The model is a simplified representation of a physical system and utilizes various approximations to describe the processes in the physical system. Model errors arise from assumed approximations and simplifications of the physical system: input parameter space has intervals in which models are valid, intervals where the fidelity of models is significantly reduced, and intervals in which the models fail. Parameter uncertainties arise because the model's input parameters are often derived from other models or experiments. The values of parameters can vary significantly, either in intervals of validity or in uncertainties of their values, and those changes have a profound effect on the model's output. Numerical errors are the third possible source of uncertainties in models. All models that involve executing simulation code during the model evaluation have additional uncertainties that arise from number round-off, discretization, coding errors, and hardware errors [104, 105].

Sensitivity analysis is focusing on quantifying the model's output dependence on input parameter uncertainties. There are many reasons to perform a sensitivity analysis, such as defining the robustness of a model to a variance of input parameters or various values of the parameters, optimizing a model output by varying parameter values, defining insensitive parameters that could be fixed in some cases [78, 104, 106]. The model's robustness is essential when the significant variance of input parameters is expected, and the model's response to those values needs to be verified. Changes in input parameters could be deterministic (the

model was designed to be valid for multiple values of input parameters) or could be caused by uncertainties in input parameters, e.g., errors in measurements from experiments or different values from different simulations. Defining the significant and insignificant parameters can help design experiments by focusing on parameters that greatly influence the model's output. Significant parameters are also a good choice for further investigations. Input parameters for sensitivity analysis are a subset of all parameters for uncertainty quantification and are direct inputs to the model investigated in the sensitivity analysis. The selection of the model's inputs and outputs depends on the model under the investigation. In the plasma discharge model, input parameters for sensitivity analysis can be: the initial values (e.g., species' densities, electron temperature, gas pressure and temperature, gas mixtures), driving electric field amplitude, frequency and pulse shape, specified reaction rates (independent of other parameters), or reaction rate coefficients. Observed output results from an investigated model depend on the goal of the model - is the primary purpose to define the dependence of time evolution or the steady-state of species' densities, electron or gas temperature, or rate of increase/decrease of selected density?

Sensitivity analysis also helps clarify the importance of individual parameters or groups of parameters in the investigated model. Parameters can be classified (ranked) into two main groups according to their importance (influence) on model output: influential and non-influential parameters. Two classification errors are possible from this classification: type I error is where the non-influential parameter is defined as important, and type II error is when the important parameter is classified as non-influential. Saltelli [78] also lists a third error, type III error, where modeling produces the correct answer for the wrong question, and the sensitivity analysis methods cannot protect against those errors. For example, type III errors are the ones when incorrect input parameter intervals (or values) are used, and the sensitivity analysis returns parameter importance for those wrong parameter values.

Depending on the analysis and methods used, the sensitivity analysis can typically be defined either as local or global. The local sensitivity analysis (LSA) focuses on defining

the variability of system responses (Y_j) on variation (perturbation) of input parameters values (x_i) around the predefined nominal value. Using the definition of the derivative, the mathematical formulation of the local sensitivity is $\frac{\partial Y_j}{\partial x_i}$ [104]. The LSA can separate input parameters into sensitive and insensitive parameters, i.e., with large and small derivatives, respectively. The derivative approach is very computationally efficient, requiring only a few evaluations of the computational model. However, it requires some changes to the existing computational model, which are not always simple to implement. Another disadvantage of the approach is that the method does not differentiate when the response is small because of insensitive parameters or non-linearity in the model itself. In other words, derivatives are descriptive around the values where they are computed and may not represent the whole input parameter space.

On the other hand, global sensitivity analysis (GSA) defines model output uncertainties to input parameter uncertainties in the whole parameter space. This approach offers the ability to explore the model's response without knowing parameter base (nominal) values and gives a more comprehensive description of the model's response. Compared to LSA, the GSA does not require any changes in a model being explored (e.g., simulation code). In general, it requires a larger number of model evaluations to explore the whole input parameter space. Using a naïve approach to the exploration of parameter space, where each parameter would take k values from the parameter interval, the total number of model evaluations grows with the number of parameters, p . With models with many parameters or relatively long evaluation time, this becomes prohibitively computationally expensive, and other approaches are required.

3.2 The Elementary Effects Method

The elementary effects (screening) method is a simple yet effective method to capture the independent effect of each parameter to the model's output, neglecting any combined effect

from multiple parameters. The elementary effects method belongs to the class of *One-factor-At-a-Time* (OAT) [107] methods that are designed to measure the importance of individual parameters by varying them individually and independently one at a time. The result of the method is the relative importance (or rank) of each parameter to the selected model output.

The fundamental idea of introducing two sensitivity measures, μ and σ , to classify model parameters according to their importance was introduced by Morris [75] in 1991. In general, a model is defined as:

$$y = f(\vec{x}) = f(x_1, x_2, \dots, x_p) \quad (3.1)$$

where x_i is the i -th input parameter of the model and p is the number of parameters. The whole input parameter space is a p -dimensional cube that is divided into k grid points (levels) in each dimension. Figure 3.1 shows an example of parameter grid for $k = 4$ grid points and for $p = 2$ parameters.

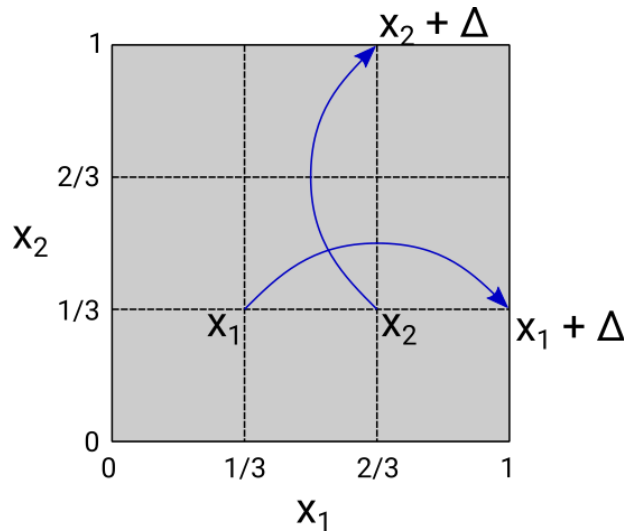


Figure 3.1 Representation of 2D grid with two parameters ($p = 2$), four grid points ($k = 4$), and parameter change $\Delta = 2/3$, used for defining parameter values for the elementary effects method. Each arrow presents points needed to compute one elementary effect: horizontal arrow to identify elementary effect for x_1 , vertical arrow for x_2 .

The elementary effect for the i -th parameter is defined as [78]:

$$EE_i = \frac{f(x_1, x_2, \dots, x_i + \Delta, \dots, x_p) - f(x_1, \dots, x_p)}{\Delta} \quad (3.2)$$

where Δ is a value from the interval $\{1/(k-1), \dots, 1-1/(k-1)\}$. When elementary effect, EE_i , is computed for each parameter at r different grid points defined by values of Δ , the two sensitivity measures are computed as [78]:

$$\begin{aligned}\mu_i &= \frac{1}{r} \sum_{j=1}^r EE_i^j \\ \mu_i^* &= \frac{1}{r} \sum_{j=1}^r |EE_i^j| \\ \sigma_i^2 &= \frac{1}{r-1} \sum_{j=1}^r (EE_i^j - \mu)^2\end{aligned}\tag{3.3}$$

Each parameter's importance and influence on the model's output can be estimated from computed values of sensitivity measures μ_i , μ_i^* , and σ , but the meaning of their absolute values depend on the model. In general, μ_i , μ_i^* present the average effect of the parameter to the model's output. When both μ_i and μ_i^* have high values, the parameter i shows a substantial effect to output, and the effect is the same (i.e., positive) in the whole parameter space. If μ_i is small (or negative) and μ_i^* is high, the parameter i has varying effects on the model's output, depending on the sampling points. The value of standard deviation σ_i indicates how sampling points affect the importance of the parameter i : the higher the σ , the more important is the choice of sampling points.

3.3 Variance-based Methods

Variance-based methods are designed to define a collective effect of input parameter uncertainties on the model's output. The parameter variance is defined by the distribution of possible values of each parameter. The distribution is selected according to the nature of the parameter. The distribution of parameter values should capture possible values and probabilities of occurrence of each value, e.g., symmetrical normal distributed values, only positive values, or uniformly distributed values in a given interval.

Variance-based methods treat the investigated model as a black box and do not rely on any model's property. When used in connection with computer simulations, variance-based models are the easiest to implement as no changes to the simulation code are required. The uncertainties in input parameters are taken into account by generating values of input parameters according to their probability distributions. Generated values of input parameters compose a list of independent simulation cases, one case for one set of input parameters, that can be run independently and will result in an ensemble of solutions from which statistical information of the model's outputs can be extracted. The whole parameter space is a p -dimensional cube, where p is the number of input parameters. Each model evaluation requires one p -dimensional point and a large number of p -dimensional points are required to explore the whole parameter space. In the case of many parameters, a large number of evaluations, or long model evaluation times, finding efficient sampling techniques is imperative to avoid excessive work required for executing variance-based methods.

3.3.1 Sensitivity Indices

Computing sensitivity indices can measure the sensitivity of model outputs due to variation of input parameters. Saltelli [78] presented sensitivity indices that enable ranking a model's parameters according to the variance of each parameter to the variance of the model's output. Using the model definition from Eq. (3.1), let the variance of outputs from all model realizations, i.e., realizations that include all values of all parameters, be labeled as $V(y)$. Let the variance of outputs from model realizations without a variance from parameter i , i.e., keeping parameter i fixed at value x_i^* , be labeled as $V_{x_i}(y|x_i = x_i^*)$, e.g., outputs along vertical dashed lines for selected value of factor F_u in Fig. 3.2. The computed variance shows the relation between variances at one selected value of parameter i and not for the whole interval of values. To compute the influence of parameter i to outputs in the whole interval of values of parameter i , the average of outputs for each value of parameter i is computed, labeled as $\mathbb{E}_{x_i}(y|x_i = x_i^*)$. The average value of model outputs is computed along vertical

dashed lines in Fig. 3.2, one average value for each vertical dashed line. The variance of computed average values define the influence of parameter i on the model outputs, labeled as $V[\mathbb{E}(y|x_i)]$.

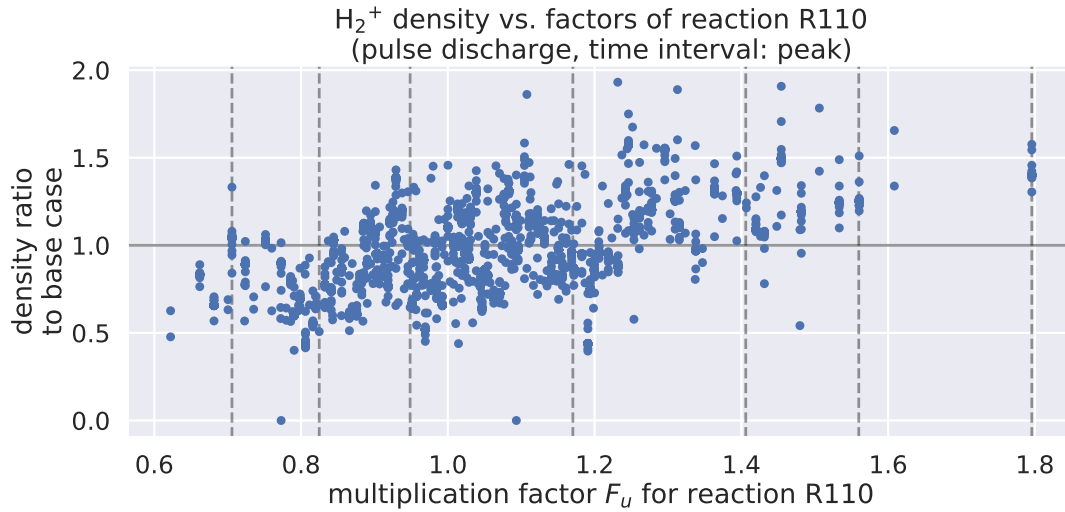


Figure 3.2 Scatter plot of model realizations (model output values) depending on values of one parameter, a multiplication factor F_u for reaction R110. Points along vertical lines have the same value for factor F_u , but different values for other factors. The use of factor F_u is defined in section 3.4.1.

The first-order indices quantify each parameter's particular importance, i.e., the importance without interaction with other parameters, and is defined for parameter i as [78]:

$$S_i = \frac{V[\mathbb{E}(y|x_i)]}{V(y)} \quad (3.4)$$

The second and higher order indices define the level of interaction between parameters. The second-order indices S_{ij} , defining interaction between parameters i and j ($i \neq j$), are defined following the same deduction as for the first order index S_i . The variance of model realizations are averaged on a plane defined by $x_i = x_i^*$ and $x_j = x_j^*$ and are labeled as $V[\mathbb{E}(y|x_i, x_j)]$. The second order sensitivity index for interaction between parameters i and j is defined as:

$$S_{ij} = \frac{V[\mathbb{E}(y|x_i, x_j)]}{V(y)} \quad (3.5)$$

The variance of model realizations when three parameters (i , j , and k) are considered is defined as $V[\mathbb{E}(y|x_i, x_j, x_k)]$ and third-order sensitivity index is defined as $S_{ijk} = \frac{V[\mathbb{E}(y|x_i, x_j, x_k)]}{V(y)}$.

The total effects sensitivity index for parameter i can be computed as [78]:

$$S_{T_i} = 1 - \frac{V[\mathbb{E}(y|x_{\sim i})]}{V(y)} \quad (3.6)$$

where $V[\mathbb{E}(y|x_{\sim i})] = V[\mathbb{E}(y|x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_p)]$ is the variance of model's outputs without parameter i .

The sensitivity indices S_i and S_{T_i} have values between 0 and 1, where smaller values indicate the parameter i is non-influential and larger values of indices indicate a more influential parameter. They define the relative importance (influence) of input parameters and give some general properties of each parameter:

- Index S_i gives the individual effect of the parameter on the model's output (i.e., only the i -th parameter without interaction with other parameters), and index S_{T_i} gives the total effect, including interaction with other parameters.
- The difference between the total and the individual effect index, $S_{T_i} - S_i$ gives the measure of how much interaction is between i -th parameter and other parameters.
- If $S_{T_i} \ll 1$ or $S_{T_i} = 0$, the i -th parameter is non-influential and can be excluded from any sensitivity analysis as it has a small or negligible effect on the variance of outputs.

3.3.2 Computing Sensitivity Indices

The main task in computing sensitivity indices is an estimation of conditional variances $V[\mathbb{E}(y|x_i)]$ as they involve computing multidimensional integrals in p -dimensional parameter space. The brute-force approach for computing sensitivity indices for one parameter would require $O(M^2)$ model evaluations, where M is the number of discrete values for each parameter. Saltelli [76] developed a method of computing sensitivity indices based solely on model evaluations, e.g., computer simulations, where the required number of computations

is significantly lower than $O(pM^2)$. With M being the number of discrete values for each parameter and p being the number of parameters, the total number of model evaluations is $M(p+2) \ll O(pM^2)$. The steps for computing sensitivity indices are the following [78]:

- generate $M \times 2p$ random parameter values from the interval of interest and split them into matrices A and B in a way that each matrix takes half of the values. The matrix A takes columns $1 \dots p$ and the matrix B takes columns $p+1 \dots 2p$, as shown in Eqs. (3.7) and (3.8):

$$A = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_i^{(1)} & \dots & x_p^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_i^{(2)} & \dots & x_p^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_1^{(M-1)} & x_2^{(M-1)} & \dots & x_i^{(M-1)} & \dots & x_p^{(M-1)} \\ x_1^{(M)} & x_2^{(M)} & \dots & x_i^{(M)} & \dots & x_p^{(M)} \end{bmatrix} \quad (3.7)$$

$$B = \begin{bmatrix} x_{p+1}^{(1)} & x_{p+2}^{(1)} & \dots & x_{p+i}^{(1)} & \dots & x_{2p}^{(1)} \\ x_{p+1}^{(2)} & x_{p+2}^{(2)} & \dots & x_{p+i}^{(2)} & \dots & x_{2p}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{p+1}^{(M-1)} & x_{p+2}^{(M-1)} & \dots & x_{p+i}^{(M-1)} & \dots & x_{2p}^{(M-1)} \\ x_{p+1}^{(M)} & x_{p+2}^{(M)} & \dots & x_{p+i}^{(M)} & \dots & x_{2p}^{(M)} \end{bmatrix} \quad (3.8)$$

- create matrices C_i that have all elements from the matrix B , except values in i -th column are taken from the matrix A :

$$C_i = \begin{bmatrix} x_{p+1}^{(1)} & x_{p+2}^{(1)} & \dots & x_i^{(1)} & \dots & x_{2p}^{(1)} \\ x_{p+1}^{(2)} & x_{p+2}^{(2)} & \dots & x_i^{(2)} & \dots & x_{2p}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{p+1}^{(M-1)} & x_{p+2}^{(M-1)} & \dots & x_i^{(M-1)} & \dots & x_{2p}^{(M-1)} \\ x_{p+1}^{(M)} & x_{p+2}^{(M)} & \dots & x_i^{(M)} & \dots & x_{2p}^{(M)} \end{bmatrix} \quad (3.9)$$

- evaluate model for all parameter values in matrices A , B and C_i and store results from evaluations into column vectors of dimension $M \times 1$:

$$y_A = f(A) \quad y_B = f(B) \quad y_{C_i} = f(C_i) \quad (3.10)$$

Using parameter values defined in matrices A , B , and C_i and model evaluations collected in vectors y_A , y_B , and y_{C_i} , the first-order sensitivity indices are estimated as follows:

$$S_i = \frac{\frac{1}{M} \sum_{j=1}^M y_A^{(j)} y_{C_i}^{(j)} - f_0^2}{\frac{1}{M} \sum_{j=1}^M \left(y_A^{(j)}\right)^2 - f_0^2} \quad (3.11)$$

where superscript (j) defines the row from vector of results (y_A , y_B , and y_C) and f_0 is the mean computed as:

$$f_0 = \left(\frac{1}{M} \sum_{j=1}^M y_A^{(j)} \right)^2 \quad (3.12)$$

The total effects sensitivity index is always greater or equal to S_i and is estimated as:

$$S_{T_i} = 1 - \frac{\frac{1}{M} \sum_{j=1}^M y_B^{(j)} y_{C_i}^{(j)} - f_0^2}{\frac{1}{M} \sum_{j=1}^M \left(y_A^{(j)}\right)^2 - f_0^2} \quad (3.13)$$

3.4 Global Sensitivity Analysis in Plasma Discharges

Global sensitivity analysis (GSA) can be used to define the system's sensitivity to uncertainty in input parameters, i.e., how changes in input parameters will affect the simulation results. Regardless of the source of the uncertainties, the parameter distribution is required to perform the GSA. GSA is a non-intrusive method utilizing the simulation code as a "black box" and performs a required number of simulation runs to define the effect of the input parameter uncertainty on quantities of interest (QoI), e.g., species densities, temperatures. The reaction rate coefficients were selected as input parameters with specified uncertainties for all global sensitivity analyses in this chapter. The variation of the reaction rate coefficients was taken into account by multiplying each reaction rate coefficient by a selected

factor (F_u) before using it in the simulation. Multiplication factors (F_u) were defined from a selected distribution. To define the base case, i.e., the case used to measure changes of output values from simulation, the multiplication factor $F_u = 1.0$ was used for all reaction rate coefficients. Using the value 1.0 also ensured that unchanged reaction rate coefficients from the reaction input file were used for a base case.

With a large number of reactions and the number of samples needed to describe the distribution of reaction rate coefficients, the total number of simulation runs using naïve approach is prohibitively large. Saltelli sequencing (an improvement of Sobol’s sequencing [76]) was used to define multiplication factors for rate coefficient and to reduce the total number of simulation runs. Using Saltelli sequencing, the total number of simulation cases is reduced from $O(p^N)$ to $O(Np)$, where N is the number of samples (factors for rate coefficients) we select to describe the rate distribution and p is the number of reaction rate coefficients (reactions in the system). In case of $p = 27$ and $N = 100$, the naïve approach would require $Np + 1 \sim 100^{27}$ cases, but Saltelli sequencing requires only $N \cdot (p + 2) + 1 = 2901$ simulation runs. Generated simulation cases are independent and were run in parallel.

3.4.1 Uncertainties of Reaction Rate Coefficients

For estimating the species density distributions, two simulation input parameters have to be defined: the distribution of the input parameters and the parameters of the selected distribution (e.g., median, variance). A normal (Gaussian) distribution is usually assumed without any knowledge of the reaction rate coefficient distribution. In our case, the normal distribution is not very convenient since factors defined by the distribution must be positive numbers, and a log-normal distribution was assumed for input parameter distribution. The second assumption was made regarding the parameters for the log-normal distribution (Fig. 3.3). Since the multiplication factors (F_u) were used to define uncertainties in the reaction rate coefficients, factor $F_u = 1.0$ corresponds to the reaction rate coefficient used in simulations without considering uncertainties. The value $F_u = 1.0$ was selected as the

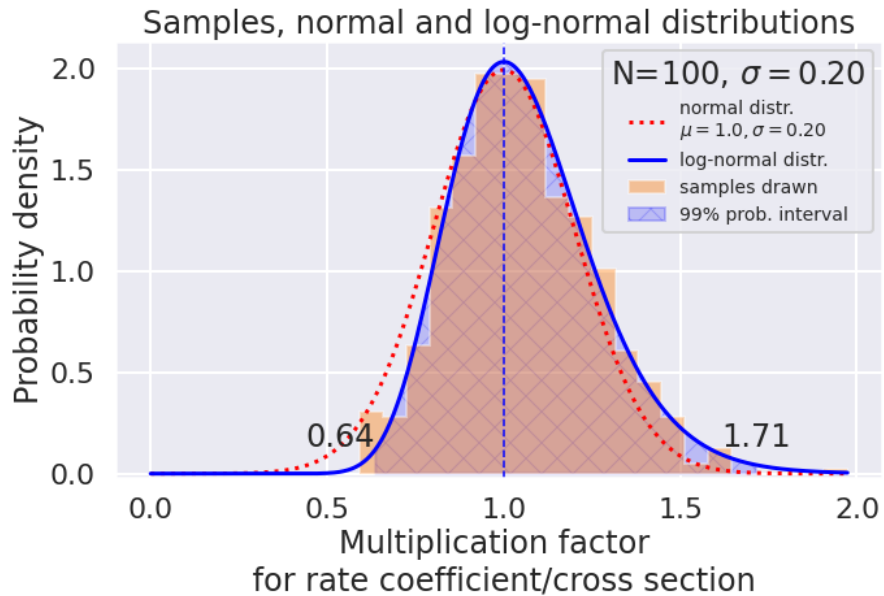


Figure 3.3 The comparison of distributions for sampling factors for reaction rate coefficient factors: log-normal distribution samples and the distribution (bar charts and solid line, respectively), and normal distribution (dotted line) with mode (the most probable value) of 1.0 and standard deviation $\sigma = 0.2$. The interval that encloses 99% of the probability of log-normal distribution is labeled with its low (0.64) and high (1.71) boundary. The most probable value of the coefficient is shown as vertical dashed line.

mode of the distribution, i.e., the most probable value returned by the distribution. The variance of the distribution was selected after gathering the uncertainties reported in the literature. When reaction rate coefficients were defined with cross-sections, the uncertainties reported in the literature were from 6% for O_2 ionization [108] to 20% for other processes. The only exception was H_2 excitation, where the uncertainty reported was 37% [109]. When reaction rate coefficients were defined using the Arrhenius form (Eq. 3.14 with $F_u = 1.0$), only uncertainties for the factor A were reported [80]. The uncertainties for A were given as a $\Delta A/A$ ratio, where ΔA is the change in A . Reported ratios for $\Delta A/A$ were between 0.05 and 5.0, but most ratios were in the range 0.3 – 0.5 (multiplication factor F_u between 1.3 and 1.5). Considering a log-normal distribution and enclosing 99% of samples from the distribution (interval inside $\pm 3\sigma$ for normal distribution), selecting $\sigma = 0.2$ gives the upper

limit for $\Delta A/A$ at 0.71 ($F_u = 1.71$, value 1.71 in Fig.3.3). The standard deviation $\sigma = 0.2$ encloses the majority of the factors reported in Ref. [80], and it was used in all simulations presented in this chapter.

The uncertainty of reaction rate coefficients was taken into account through multiplication factor F_u in two different ways, depending on how the reaction rate coefficient was defined:

- as an analytical function, e.g., as the Arrhenius form:

$$K(T) = F_u \cdot A \cdot T^B \cdot \exp\left(-\frac{C}{T}\right) \quad (3.14)$$

where T is the temperature of the incident species (either electron or gas temperature), and A , B , and C are the unchanged coefficients for given reaction rate

- as a cross section for electron impact reactions:

$$K(\sigma, f) = \int_0^\infty \sqrt{\frac{2e\varepsilon}{m_e}} F_u \sigma(\varepsilon) f(\varepsilon) d\varepsilon \quad (3.15)$$

where e is the elementary charge, m_e is the electron mass, $\sigma_i(\varepsilon)$ is the reaction cross section, $f(\varepsilon)$ is the electron energy distribution function (EEDF) and ε is the electron energy.

The factor F_u was drawn from a log-normal distribution to incorporate the uncertainties in reaction rate coefficients. The uncertainties for all reaction rate coefficients, either defined with a cross-section or analytical function, were the same and had a standard deviation $\sigma = 0.2$. The mode of the distribution, i.e., the most probable value according to the selected log-normal distribution, was selected to be 1.0, corresponding to reaction rate coefficients defined in simulations without considering the uncertainties.

3.4.2 Plasma Discharge in Argon

The Global Sensitivity Analysis (GSA) was performed in a pure argon plasma discharge with constant absorbed power. Argon was selected as a background gas because the reaction

rate coefficients for the ground state, ionization, and first excited state are well known and investigated. A cylindrical system with radius $R = 0.5641$ cm and length $L = 1$ cm (volume $V = 1$ cm³) was used, background gas pressure $p = 760$ Torr and temperature $T_g = 300$ K were both kept constant. The initial effective electron temperature $T_e = 2$ eV and was computed with coupled Boltzmann equation (BE) solver. The deposited power, $P_{abs} = 50$ W, was kept constant during the simulation. The electron energy distribution functions (EEDFs) were evaluated with a coupled BE solver on a 1% change of the reduced electric field. The simulation was run up to 1 ms when the system reached the steady-state. Quasi-neutrality was not enforced, and electrons were treated as a separate species. Species in the system were electrons (e), ground state argon atoms (Ar), atomic argon ions (Ar^+), molecular argon ions (Ar_2^+), excited argon atoms were lumped into a single excited state (Ar^*). The vibrational and rotational states of molecular argon ions (Ar_2) were neglected. The reaction set was composed of ten reactions, and they are listed in Table 3.1. Reaction rate coefficients for reactions R1-R4 were defined as cross-sections, and the other rate coefficients were defined as analytical functions.

Table 3.1 Reactions used in the plasma discharge in argon with their reaction rate coefficients and threshold energies. *Note:* T_e in eV, T_g in K.

Reaction	Rate coefficient, unit	ΔE_i [eV]	Refs.
R1: $\text{Ar} + e \Rightarrow \text{Ar} + e$	cross section		[90]
R2: $e + \text{Ar} \Rightarrow 2e + \text{Ar}^+$	cross section	15.76	[90]
R3: $e + \text{Ar} \Rightarrow \text{Ar}^* + e$	cross section	11.50	[90]
R4: $e + \text{Ar}^* \Rightarrow \text{Ar}^+ + 2e$	cross section	4.30	[90]
R5: $\text{Ar}_2^+ + e \Rightarrow \text{Ar}^* + \text{Ar}$	$8.5 \times 10^{-13} (11600 T_e / 300.0)^{-0.67}$, m ³ /s	$1.5 T_e$	[44]
R6: $\text{Ar}_2^+ + \text{Ar} \Rightarrow \text{Ar}^+ + 2\text{Ar}$	$6.06 \times 10^{-12} / T_g \exp(-15130.0 / T_g)$, m ³ /s		[44]
R7: $\text{Ar}^* + \text{Ar}^* \Rightarrow \text{Ar}_2^+ + e$	6.0×10^{-16} , m ³ /s	$-1.5 T_e$	[44]
R8: $\text{Ar}^+ + 2e \Rightarrow \text{Ar} + e$	$8.75 \times 10^{-39} T_e^{-4.5}$, m ³ /s	$1.5 T_e$	[44]
R9: $\text{Ar}^* + 2\text{Ar} \Rightarrow \text{Ar} + 2\text{Ar}$	1.4×10^{-44} , m ³ /s		[44]
R10: $\text{Ar}^+ + 2\text{Ar} \Rightarrow \text{Ar}_2^+ + \text{Ar}$	$2.25 \times 10^{-43} (T_g / 300.0)^{-0.4}$, m ⁶ /s		[44]

3.4.2.1 GSA Parameters

The input parameters for GSA were multiplication factors for reaction rate coefficients or cross-sections, distributed according to the log-normal distribution. All input parameters were independently sampled from the same distribution: the distribution mean was 1.0, the standard deviation $\sigma = 0.2$, and 100 samples were drawn for each parameter. Plots and histograms in Fig. 3.4 show the initial distribution (dashed line) and the distribution of samples (histogram) for two reactions: (a) elastic scattering reaction R1, and (b) ionization reaction R2. Quantities of interest (QoI) were densities of all species, except ground state argon.

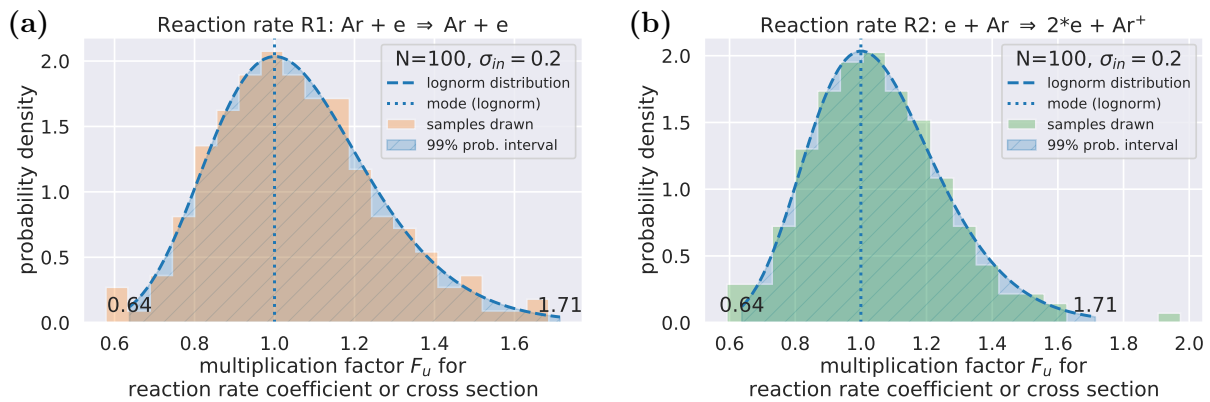


Figure 3.4 The initial distributions of reaction rate coefficient factors for (a) elastic scattering (reaction R1) and (b) ionization (reaction R2) with $N = 100$ samples. The dashed line shows the log-normal distribution, and the histogram shows the distribution of samples. The vertical dash-dotted line shows the distribution mode (the most probable value of the distribution).

3.4.2.2 GSA Results

Output values from the simulation were species' densities collected in the last 200 time steps when the system was already in a steady state. Figure 3.5 shows the time evolution of species densities, electron temperature, and the time interval used for GSA (shaded area). The general sensitivity of the system can be investigated by comparing the distributions

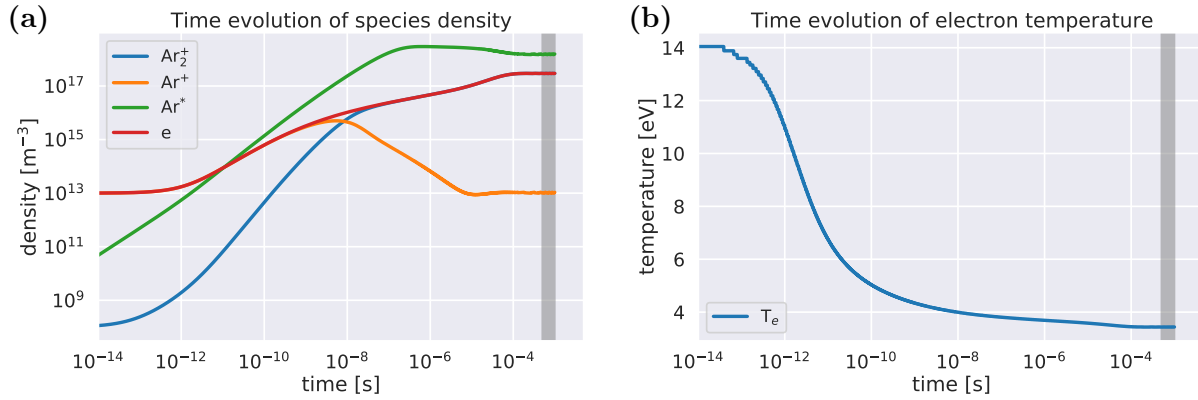


Figure 3.5 Time evolution of densities (a) and electron temperature (b) in the simulation case (base case) without uncertainties in the reaction rate coefficients. Results acquired in the last 200 time steps of the simulation (shaded area on the plot) were used in a global sensitivity analysis.

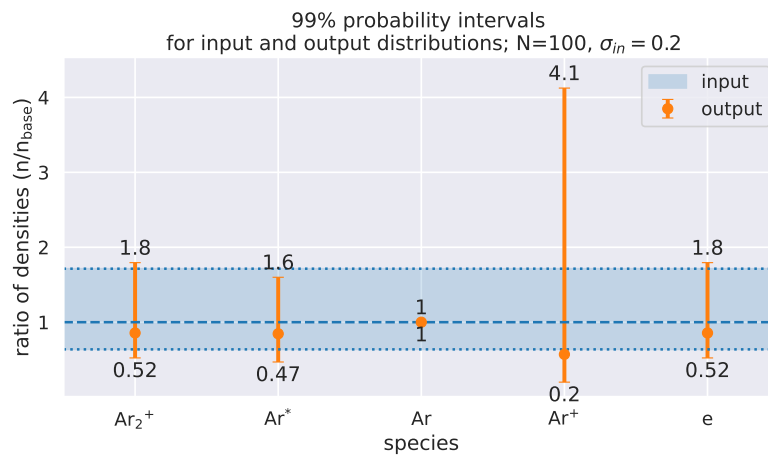


Figure 3.6 Comparison of density mode and intervals that are enclosing 99% of all densities of given species. Reaction rate coefficient distributions had mode 1.0, $\sigma = 0.2$, and were sampled with $N = 100$ samples.

of input parameters and simulation results. As mentioned before, reaction rate coefficients (input parameters) had deviation $\sigma = 0.2$. Figure 3.6 shows the 99% probability intervals for all species in the system. The probability intervals were computed from species' densities (simulation results) by assuming the log-normal distribution and fitting its parameters to the simulation results. The results show that argon ions are the most sensitive to the input

uncertainties. Special care needs to be given to selecting rate coefficients if the quantity of interest is the density of atomic ions. On the contrary, input parameter uncertainties have almost no effect on ground state argon density. Results also show that molecular ions and electrons have the same sensitivity to input uncertainties.

The distributions of selected species are presented as histograms in Fig. 3.7. The comparison of log-normal distributions and histograms shows good agreement and confirms that a log-normal distribution was a reasonable assumption for describing the distribution of output densities. The log-normal parameters were acquired from fitting the log-normal distribution to the simulation results, and 99% probability intervals for all densities were computed.

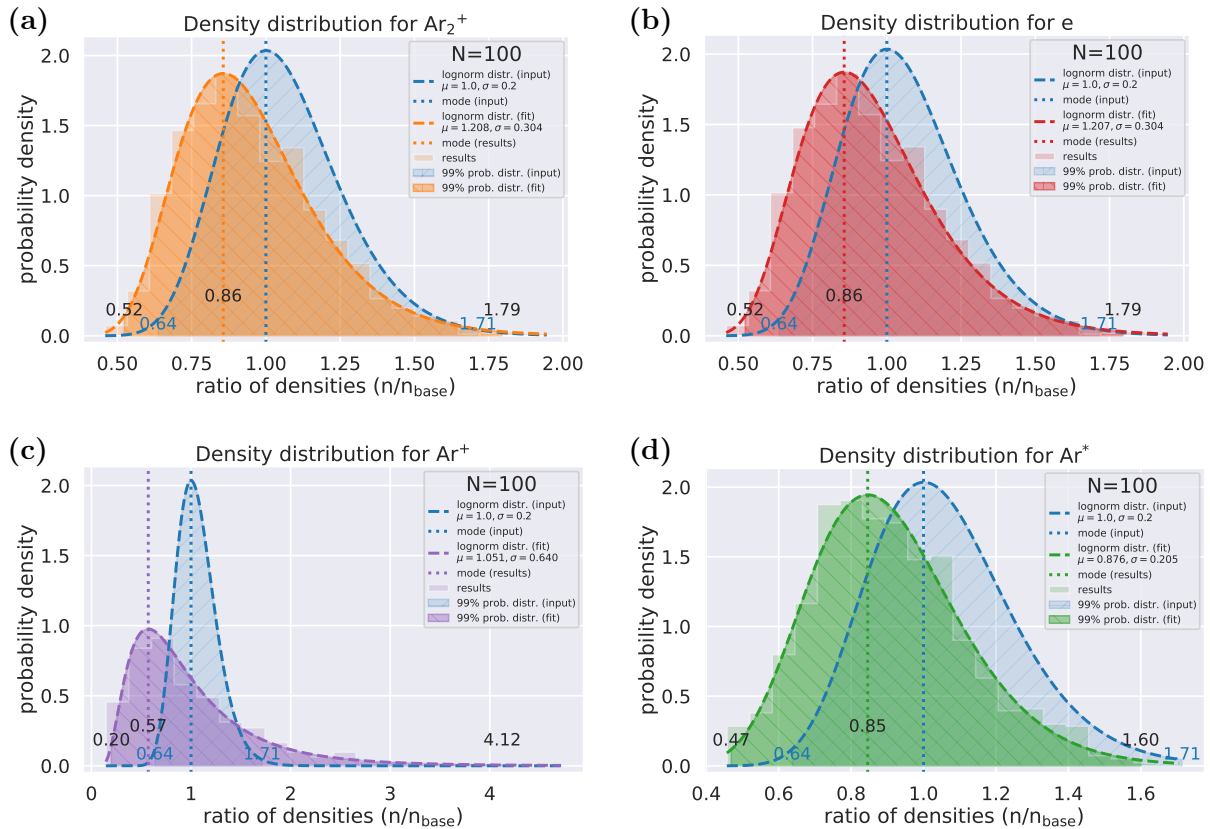


Figure 3.7 The distribution of species densities (results of the simulation) of selected species: (a) molecular ions Ar₂⁺, (b) electrons, (c) atomic ions Ar⁺, and (d) excited species Ar^{*}. Densities from results are shown as histograms. The distributions fitted to results and the input distributions are shown as dashed lines. The vertical dotted line marks the most probable values from each distribution.

The results in Fig. 3.7 show that the most probable species density (vertical dotted line) varies slightly between input and output distributions. The output distributions have, in most cases, larger deviation compared to the input distributions, except for ground state argon (see Fig. 3.6) where the density is not sensitive to input uncertainties. Shaded areas below each distribution curve mark the 99% probability interval for input and output distributions. The numbers next to the distribution curves show upper and lower limits of probability interval for each density.

Figures 3.8 and 3.9 show a relative importance of individual reactions on species densities using the first-order (S_i) and the total (S_{T_i}) sensitivity indices. It is important to keep in mind that GSA indices are showing a *relative importance* of the reaction rate coefficient uncertainty on the output density and not the importance of the reaction itself. The reaction importance can be defined through reaction set reduction (e.g., using the directed relation graph (DRG) method). The individual reaction could be unimportant, but the variance on the reaction rate coefficient could have a significant relative effect on output densities.

Figure 3.8(a) shows reaction importance on Ar_2^+ density. Both sensitivity indices rank the same reactions as important: R1, R4, R9, R5, and R7. Other reactions are not important, having sensitivity indices equal to zero. The difference between S_{T_i} and S_i is the slight interaction of reactions R1 (elastic scattering) and R4 (step-wise ionization) with other reactions. The reactions R9, R5, and R7 do not interact with other reactions since their indices S_{T_i} and S_i are practically the same. Figure 3.8(b) shows reaction importance on electron density. Comparing reaction importance for Ar_2^+ and electron density shows that the order of reaction importance is the same for both species, which is not surprising because the densities used in the GSA for computing reaction importance are taken from the steady-state system when densities of both species are the same (see Fig. 3.5).

Figure 3.9(a) shows reaction importance on Ar^+ density, showing similar reaction order (ranking) as for Ar_2^+ and electrons. Elastic scattering (R1) and step-wise ionizations (R4) are the most important reactions, followed by argon de-excitation (R9) and atom-assisted

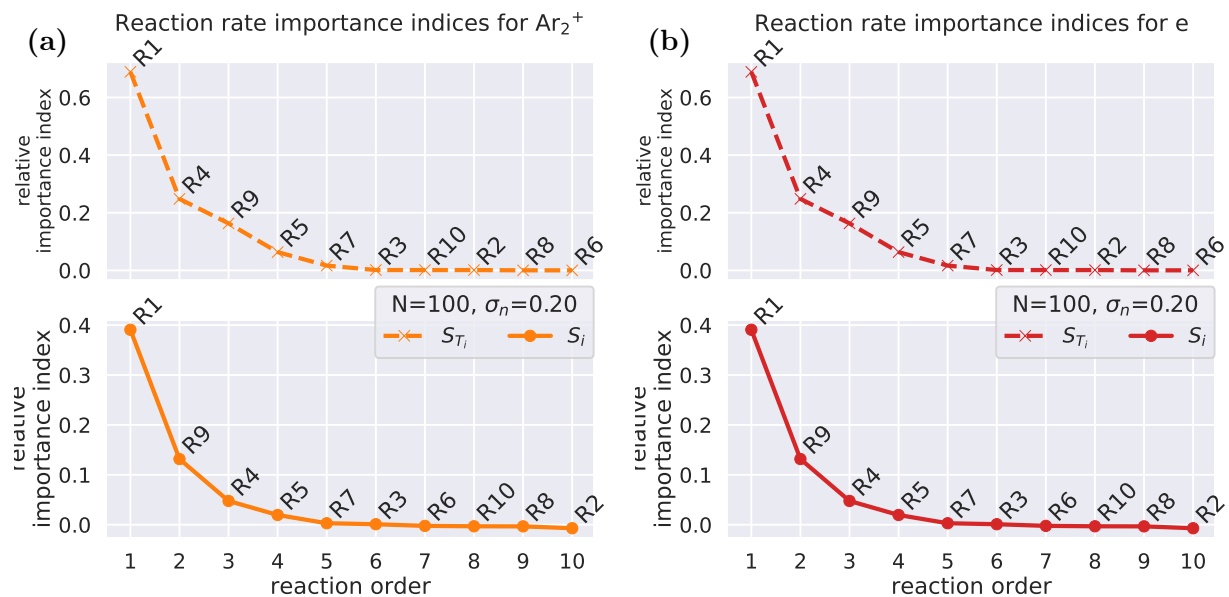


Figure 3.8 Relative importance of reactions on species' densities for (a) molecular ions Ar_2^+ and (b) electrons. The relative importance was defined according to the sensitivity indices S_{T_i} and S_i .

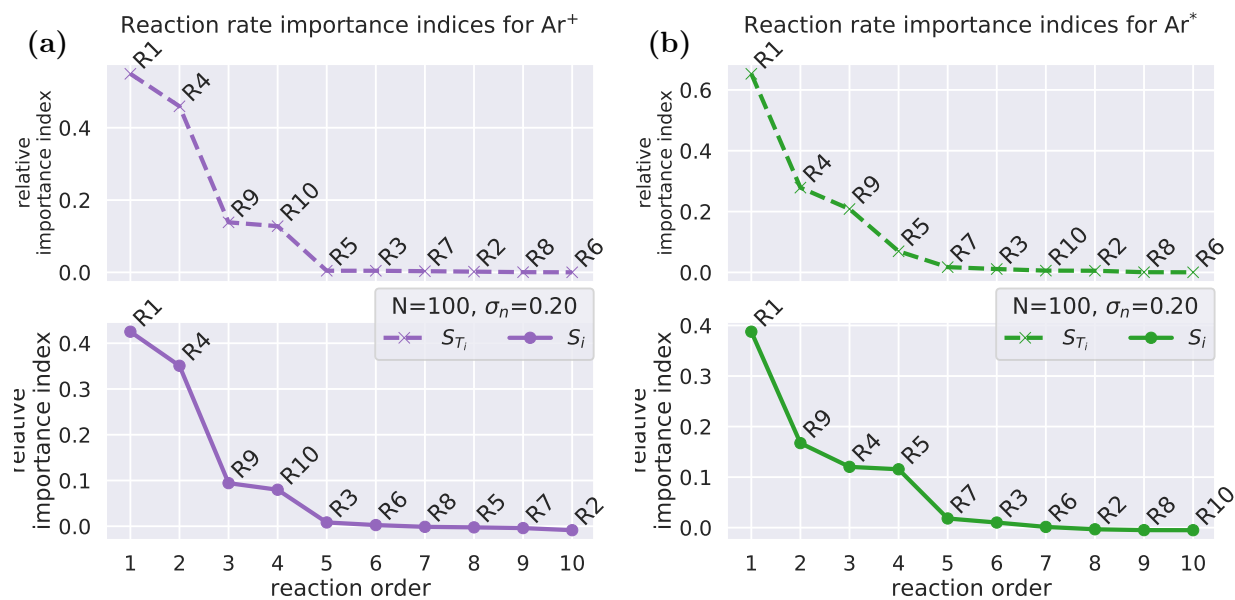


Figure 3.9 The relative importance of reactions on species' densities for (a) atomic ions Ar^+ and (b) excited species Ar^* . The relative importance was defined according to sensitivity indices S_{T_i} and S_i .

association (R10). Reaction importance on excited argon (Ar^*) is presented in Figures 3.9(b), showing similar order of reactions as for the other species. Reactions with small threshold energies, i.e., reactions R1, R4, and R9, dominate at high pressure.

3.4.2.3 Conclusion

Global sensitivity analysis (GSA) of plasma discharges in pure argon at atmospheric pressure was performed in a reaction set with a single excited state species, atomic and molecular ions, ground state argon, and electrons. The results from the GSA can be interpreted from two different viewpoints: prioritizing reaction sensitivity based on the effect they have on species' densities and which species is the most sensitive to the uncertainties in the reaction rate coefficients.

Sensitivity indices S_{T_i} and S_i show that uncertainties in reaction rate coefficients for elastic scattering (reaction R1), step-wise ionization (reaction R4), and de-excitation (reaction R9) reactions have the largest effect on species' densities at the steady-state in high pressure ($p = 760$ Torr). The elastic scattering (R1) has a strong role in shaping the EEDF at high pressure, and its role as the dominant reaction in high pressure is also shown by sensitivity indices S_{T_i} and S_i . Uncertainties in reaction rate coefficients for reactions R2 (ionization), R6 (dissociation by atom impact), R8 (recombination), and R10 (atom assisted association) show a minor influence on densities at the steady-state. The output densities show that log-normal distribution can be assumed for the output densities (Fig. 3.7). The distributions of input parameters and output densities can be compared using distribution mode and standard deviation. The Ar^+ density is the most sensitive to the uncertainties in reaction rate coefficients, with the distribution mode=0.85 (input distribution mode=1.0) and deviation $\sigma = 0.64$ (input $\sigma = 0.2$). The high sensitivity of Ar^+ density can be observed in Fig. 3.6, where ratios of output densities are presented. The 99% probability interval for Ar^+ density is much wider, 0.2 – 4.12, compared to the input distribution probability interval, 0.64 – 1.71. The 99% probability intervals for Ar_2^+ , Ar^* , and electron densities, all inside

0.42 – 1.8 and are very similar to the input probability interval (0.64 – 1.71). The density of ground state argon shows no sensitivity to uncertainties in the reaction rate coefficients, which can be attributed to the partially ionized ($\sim 10^{-4}$) discharge.

3.4.3 Plasma Discharge in H₂-O₂-Ar Mixture

The global sensitivity analysis of plasma discharges was performed in a flow reactor with volume $V = 5 \text{ cm}^3$, gas mixture with 1% H₂ - 0.15% O₂ - 98.85% Ar, and inflow and outflow velocities $v = 8.5 \text{ m/s}$. Pressure and gas temperature were kept constant at $p = 300 \text{ Torr}$ and $T_g = 500 \text{ K}$, respectively. The electron temperature T_e was computed using the electron energy balance equation. Sheath and losses to the wall were neglected. Quasi-neutrality was not enforced (electrons were treated as a separate species), and the electron energy distribution function (EEDF) was assumed to be Maxwellian but electron temperature dependent. The reaction set was composed of 55 species (Tab. 3.2) and 146 reactions (Tab. C.1 in Section C).

Table 3.2 Species used in the reaction set for 250 ns plasma discharge in H₂-O₂-Ar gas mixture with 1% H₂, 0.15% O₂, and 98.85% Ar. The reaction set is presented in Tab. C.1 (section C).

ground state	excited	charged
Ar	Ar _{e=1,2}	Ar ⁺ , Ar ₂ ⁺
H, H ₂ , H ₂ O	H _{e=1,2} , H _{2,r=1,2} , H _{2,v=1-3} , H _{2,e=1-10}	H ⁺ , H ₂ ⁺
HO ₂ , H ₂ O ₂	H ₂ O _{v=1,2}	H ₂ O ⁺
O, OH, O ₂ , O ₃	O(¹ S), O(¹ D), O _{2,r=1} , O _{2,v=1-4} , O ₂ (a ¹ Δ), O ₂ (b ¹ Σ), O _{2,e=3-7}	O ⁺ , O ⁻ , O ₂ ⁺ , O ₂ ⁻ , O ₃ ⁺

The power was deposited using pulse with duration 250 ns and the repetition frequency $\nu = 20$ kHz (time between pulses $\tau = 5 \times 10^{-5}$ s) and with shape presented in Fig. 3.10. The shape of the pulse follows the experimentally measured power profile in Ref. [110], but taking only the positive part of the pulse and depositing the same amount of energy in every pulse as in experiments (2.6mJ/pulse). The simulations were run for two pulses with an end time $t = 100\mu s$. The results for performing the global sensitivity analysis (GSA) were collected during two time intervals during the simulation: the peak of the second pulse (between 50.24 and 50.5 μs) and the end of the second pulse (between 98 and 100 μs), also marked in Fig. 3.12 with vertical gray areas.

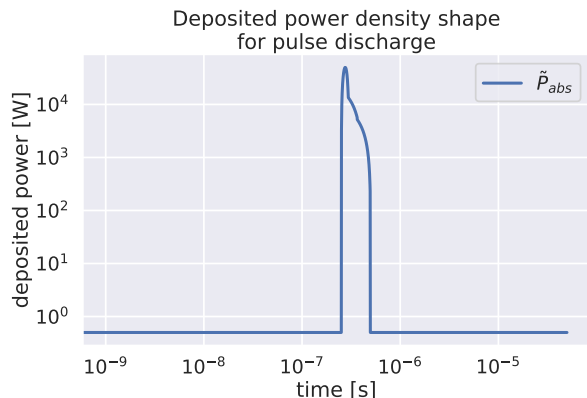


Figure 3.10 The shape of the deposited power in pulse discharge in 1% H₂, 0.15% O₂, and 98.85% Ar gas mixture. The pulse was taken from experiments shown in Ref. [110] with the repetition frequency $\nu = 20$ kHz ($\tau = 50\mu s$) and pulse duration 250 ns. The shape of the pulse follows an experimentally measured power profile, but taking only the positive part of the pulse and depositing the same amount of energy in every pulse as in the experiment (2.6mJ/pulse).

3.4.3.1 GSA Parameters

The input uncertainties were applied to reaction rates by selecting multiplication factor F_u for reaction rate coefficient (Eq. (3.14)) and cross sections (Eq. (3.15)). The values for factors F_u were independently drawn from a log-normal distribution with the distribution mode=1.0 (the most probable outcome of the distribution) and the standard deviation

$\sigma = 0.2$. Figure 3.11 shows the distribution of multiplication factor F_u for reaction R110, a histogram depicting the values drawn for F_u with dashed line presenting defined log-normal distribution. The 99% probability interval is presented with the lower (0.63) and the upper (1.71) boundaries.

To perform the global sensitivity analysis using Saltelli's sampling, $M = 100$ samples were drawn from log-normal distribution for each factor F_u . With number of reactions $p = 146$ and $M = 100$ samples drawn for each factor, the total number of simulations run was $M(p + 2) + 1 = 100(146 + 2) + 1 = 14801$.

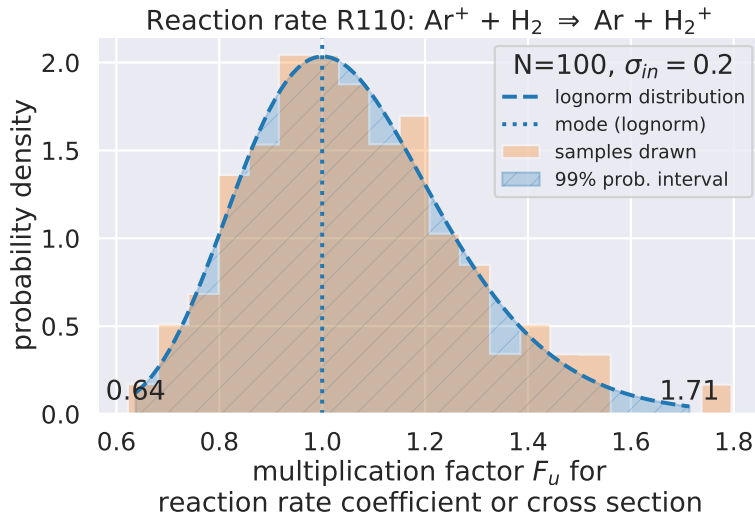


Figure 3.11 The histogram of $N = 100$ samples drawn from the log-normal distribution with the dashed line showing the initial log-normal distribution. The figure shows the multiplication factor distribution for reaction 110: $\text{Ar}^+ + \text{H}_2 \rightarrow \text{Ar} + \text{H}_2^+$.

3.4.3.2 GSA Results - output distributions

One aspect of global sensitivity analysis is defining the effect of uncertainties in input parameters on output distributions. During the discharge, the simulation results for GSA were collected at two time intervals: (a) at the peak of the second pulse (time between 50.24 and 50.5 μs), (b) at the end of the second pulse (time between 98 and 100 μs). Figure 3.12 shows

the time evolution of selected species' densities and times intervals for collecting values. The intervals for collecting results were selected to capture the effect of reaction uncertainties at the peak of the deposited power and at the end of the relaxation time before the next pulse. The average value of collected results in each interval was computed and used in the sensitivity analysis to eliminate small fluctuations in collected values.

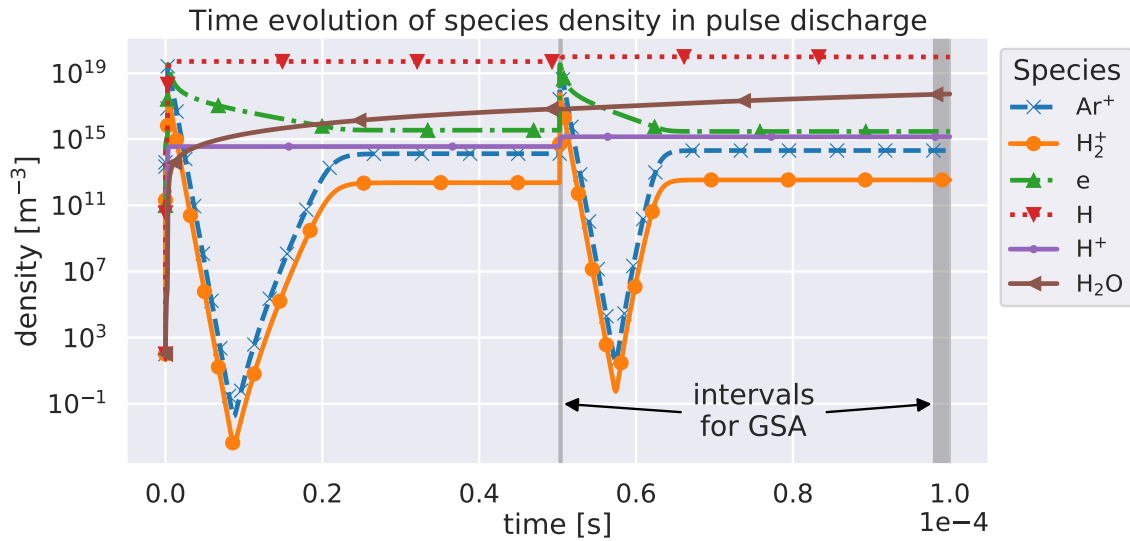


Figure 3.12 Time evolution of selected species in 250 ns pulse discharge in $\text{H}_2\text{-O}_2\text{-Ar}$ gas mixture. The light gray shaded areas mark the time intervals at which values for GSA were recorded. The GSA was made at two time intervals: (a) at the peak of the pulse (time between 50.24 and $50.5\mu\text{s}$), and (b) at the plateau before the next pulse (between 98 and $100\mu\text{s}$).

The results from all GSA simulation cases (14801 in total) were collected, and histograms were generated. The output distribution parameters were computed from generated histograms of collected results by fitting the log-normal distribution parameters to the collected results (i.e., densities or temperature). The 99% probability intervals were computed from fitted distributions for selected results at both time intervals: Fig. 3.13(a) at the pulse peak, and Fig. 3.13(b) at the end of the relaxation time before the next pulse. The species density and temperature responses to input uncertainties are very different at both collected time intervals.

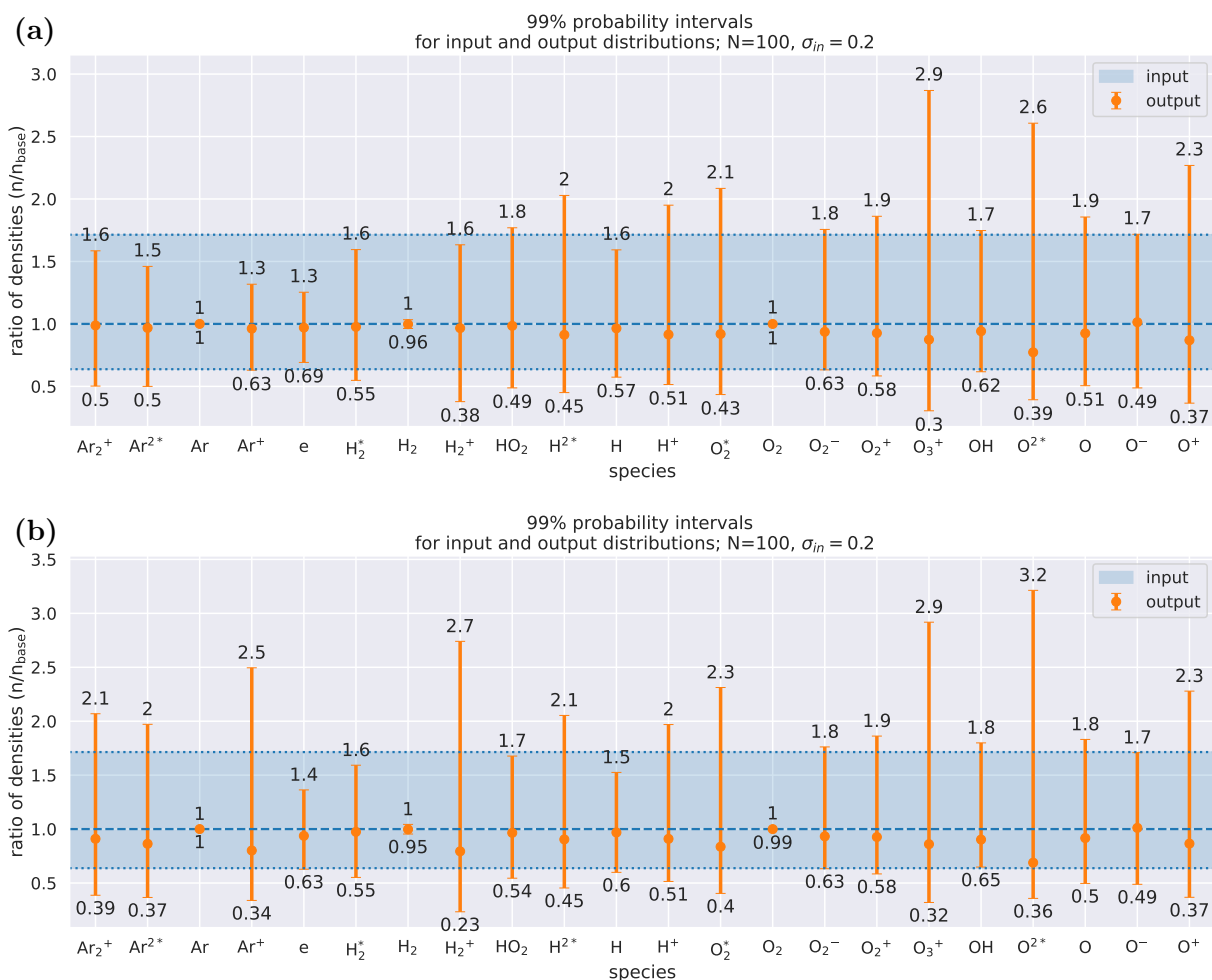


Figure 3.13 Comparison of density distribution mode and intervals that are enclosing 99% of all density values for selected species at (a) the peak of the pulse (time between 50.24 and 50.5 μs), and (b) the plateau before the next pulse (between 98 and 100 μs). All reaction rate coefficient distributions had mode=1.0 and $\sigma = 0.2$.

The plots in Fig. 3.13 show an overview of probability intervals of selected species in both time intervals. The probability intervals were computed from log-normal distributions fitted to resulting species' densities. The probability intervals for argon (charged and excited species) and hydrogen (charged) species are larger at the relaxation time (time between 98 and 100 μs in Fig. 3.13(b)) compared to the probability intervals at the peak of the pulse. The feedstock ground gas species (i.e., Ar, H_2 , and O_2) and electron temperature (T_e) show no sensitivity to the reaction rate uncertainties at either time of interest.

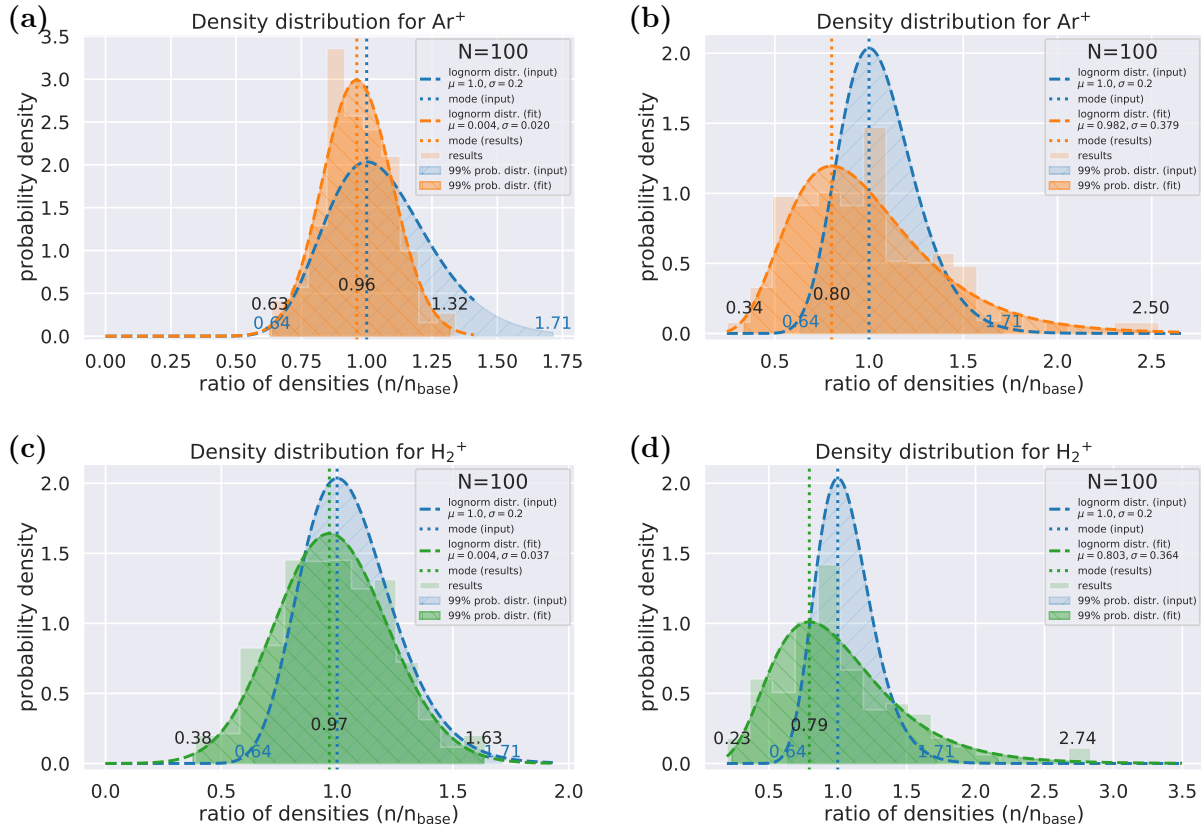


Figure 3.14 The distributions of species densities (results of the simulation) of selected species: (a) at pulse peak and (b) at relaxation time for the atomic ions Ar^+ , and (c) at pulse peak and (d) at the relaxation time for the molecular hydrogen ions H_2^+ . The species densities are shown as histograms, the log-normal distributions fitted to the results, and the input distributions are shown as dashed lines.

The individual species' density distributions reveal very different sensitivity to the input uncertainties at each time interval. Atomic argon ion (Ar^+) and molecular hydrogen ion (H_2^+) distributions show smaller sensitivity to uncertainties at the pulse peak (at the first time interval) compared to the time of relaxation (the second time interval), as shown in Fig. 3.14. The interval increases 2.8 times for argon ions (from $[0.6 - 1.35]$ to $[0.34 - 2.49]$) and 1.9 times for hydrogen molecule ions (from $[0.35 - 1.65]$ to $[0.23 - 2.73]$). The mode of Ar^+ and H_2^+ density distributions is also shifted towards smaller values at the plateau, indicating that the uncertainties will generally cause lower species' densities but also more extreme densities in certain cases. Additional analysis of sensitivity results is necessary

to find the reaction rate coefficients and their values that result in up to 2.5 times higher density of Ar^+ and 2.73 times higher density of H_2^+ . These phenomena are expected since the deposited power at the pulse peak causes electrons to have fewer collisions and have a reduced diffusion rate, resulting in a narrow electron density distribution at the peak of the pulse and wider at the plateau before the next pulse.

Similar behavior of simulation result distributions, i.e., preserving log-normal distribution and having a smaller deviation at the pulse peaks than deviation at the plateau before the next pulse, can be observed for electron density and electron temperature. The electron density at the pulse peak (Fig. 3.15(a)) shows smaller sensitivity on input uncertainties, the upper limit of 99% probability interval at pulse peak being 1.25 compared to 1.71 in the input distribution. The sensitivity after the pulse (Fig. 3.15(b)) increases, and the upper limit of 99% probability interval increases slightly, to 1.36.

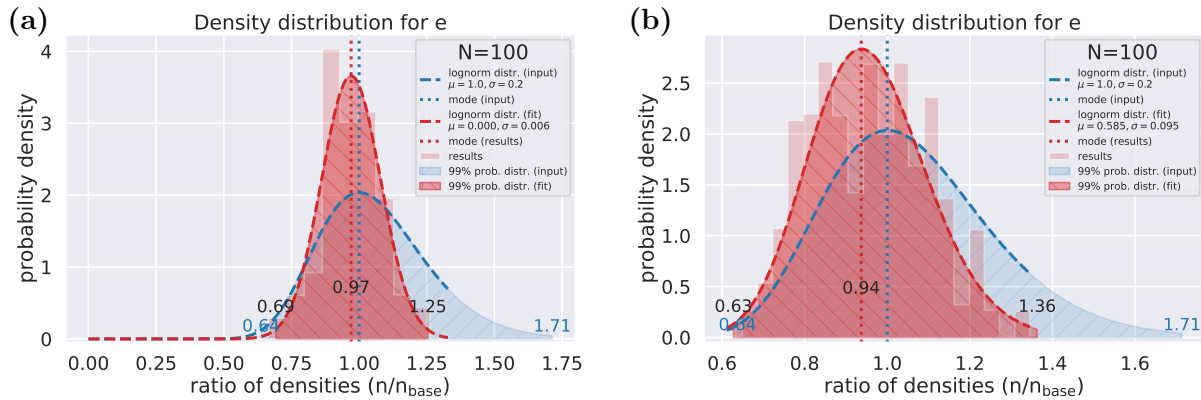


Figure 3.15 The distribution of electrons density at (a) the peak of the pulse (time between 50.24 and 50.5 μs), and (b) the plateau before the next pulse (between 98 and 100 μs). The density values are shown as histograms and the distributions as dashed lines.

The electron temperature shows minimal sensitivity to input uncertainties, as shown in Figs. 3.16(a) and (b). All values are inside the -7% and +11% interval of the T_e without considering the uncertainties. Smaller dependence of T_e on input uncertainties can be explained by looking at the reactions affecting the electron temperature. Only reactions with defined threshold energies add terms to the energy equation and directly contribute to the

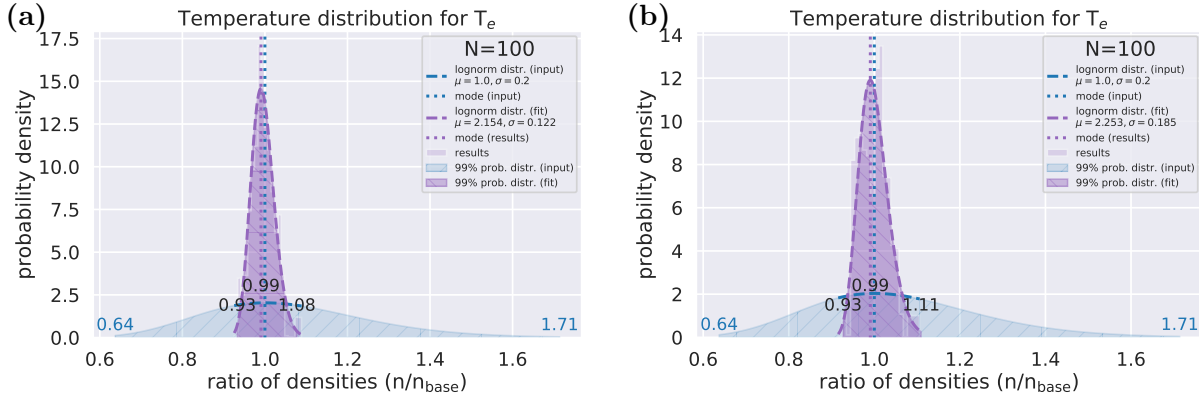


Figure 3.16 The distribution of electron temperature T_e at (a) the peak of the pulse (time between 50.24 and 50.5 μs), and (b) the plateau before the next pulse (between 98 and 100 μs). The temperature values are shown as histograms and log-normal distributions as dashed lines.

uncertainty of electron temperature through uncertainties in their reaction rate coefficients. In the reaction set used for the sensitivity analysis, 59 reactions out of 146 have the threshold energy defined. Not all species and reactions have the same contribution, and the second term in Eq. (2.2) can be separated into the contribution from different species, namely, from the feedstock gas species and other species:

$$\frac{d}{dt} \left[\frac{3}{2} n_e T_e \right] = \tilde{P}_{\text{abs}} - n_e \sum_i n_i K_i \Delta E_i - n_e \sum_j n_j K_j \Delta E_j \quad (3.16)$$

where $\tilde{P}_{\text{abs}} = P_{\text{abs}}/V$ is the absorbed power density, i is the index going over all feedstock gases (i.e., Ar, H₂, and O₂), and j is the index going over all other species. The deposited power density (\tilde{P}_{abs}) doesn't have any uncertainty and is considered as fixed from the sensitivity point of view.

The densities of feedstock gases in the second term (n_i) are much higher, by an approximate factor of 10^4 , compared to the densities of other species gathered in the third term. The second term is the dominant term for computing uncertainties in the electron temperature, and the third term can be neglected. Argon is the dominant species considering densities and reaction rate coefficients for Ar, H₂, and O₂ species. When computing the contribution from individual species and reactions, argon reactions contribute around 99% to the second

term and temperature changes. The deposited power (\tilde{P}_{abs}) term and the reaction term (term 2) are defining the value of T_e , and only uncertainties in reactions with ground argon as a reactant affect the uncertainties in electron temperature.

3.4.3.3 GSA Results - reaction importance

The second aspect of global sensitivity analysis is to define the reaction importance on selected species' density. Figure 3.17 shows the relative importance of individual reactions on H_2^+ density at both selected time intervals. The reaction importance was defined by the first-order (S_i) and the total (S_{T_i}) sensitivity indices. The sensitivity indices show a *relative importance* of the reaction rate coefficient uncertainty on the output density and not the importance of the reaction itself. Reactions R110 and R111 are the most influential ones, but at the same time, they are the opposite reactions: product species of one reaction are reactant species of the other. With the reaction rate coefficients being constant and $K_{110} \sim 1.85 \times K_{111}$, the ratio of reactant's densities defines the dominant reaction. The ratio of densities of reactants is about the same (ground state Ar to H_2 and Ar^+ to H_2^+), and the reaction R110 is the dominant reaction. Other reactions are non-influential and are

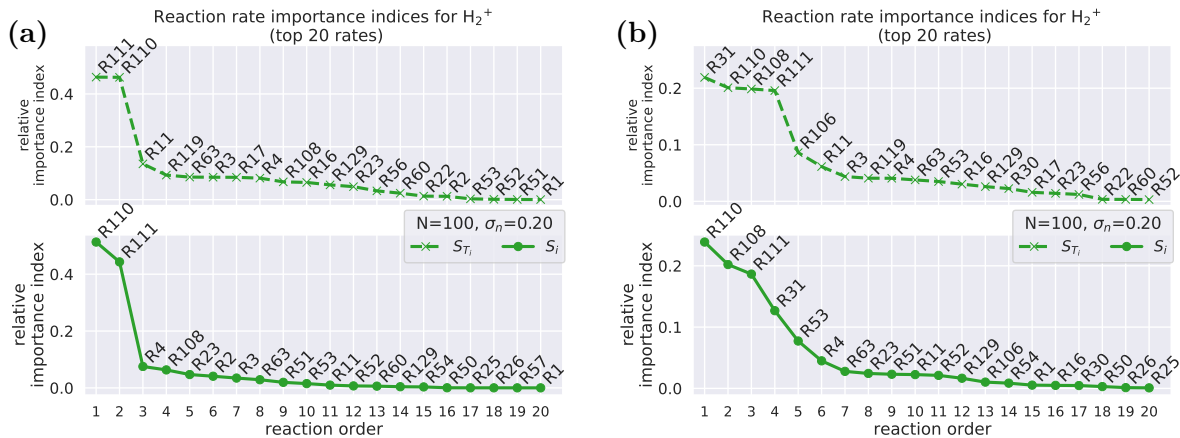


Figure 3.17 The relative importance of reactions on H_2^+ density at (a) the peak of the pulse (time between 50.24 and 50.5 μs), and (b) the plateau before the next pulse (between 98 and 100 μs). The relative importance was defined with sensitivity indices S_{T_i} and S_i .

only affecting the H_2^+ density through proxy species.

The reaction importance in terms of electron density at the pulse peak ($t_1 = 50.24 - 50.5\mu\text{s}$) is presented in Fig. 3.18 and reactions are listed in Table. 3.3. The sensitivity index S_i (individual importance) shows that argon ionization (reaction R4) is the most important reaction (bottom plot in Fig. 3.18). This is expected since the reaction produces electrons, and argon density is about 100 times larger than the O_2 and H_2 densities, acting as reactants in other reactions.

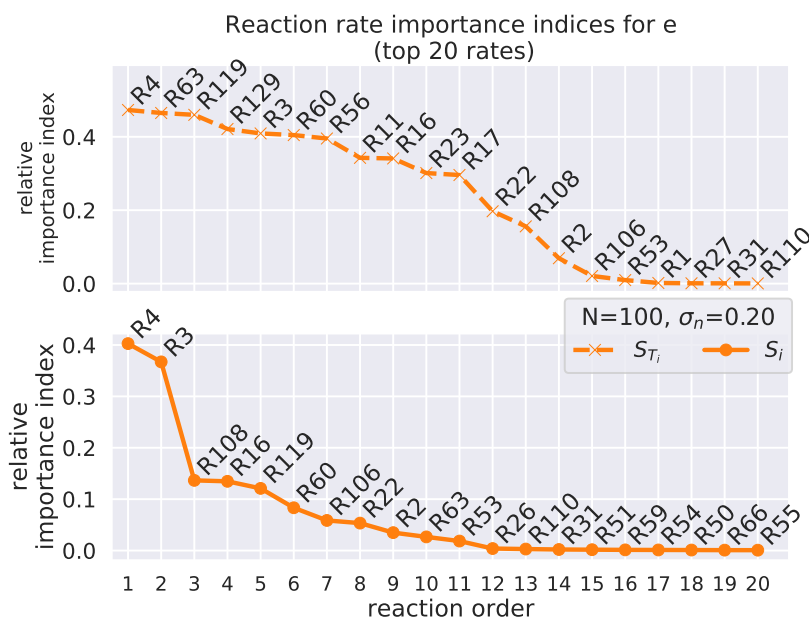


Figure 3.18 The relative importance of reactions on electron density at the pulse peak (time between 50.24 and 50.5 μs). The relative importance was defined with sensitivity indices S_{T_i} and S_i .

The top plot in Fig. 3.18 shows group of reactions (Table. 3.3) have equal importance to electron density, and that reaction R4 is the only one that modifies the electron density directly. Other reactions affect electron density indirectly via changing energy profiles. Another interesting fact is that reactions R119 and R129 ranked as third and fourth in Tab. 3.3, are reactions that do not involve electrons at all. They indirectly affect the electron density, showing that assuming the reaction importance based on species and reaction rate coefficients can lead to wrong conclusions.

Table 3.3 The most important reactions for electron density at the pulse peak (time between 50.24 and 50.5 μ s) according to the total sensitivity index S_{T_i} .

Rank	Reaction
1	R4 $\text{Ar} + e \Rightarrow \text{Ar}^+ + 2e$
2	R63 $\text{H}_2 + e \Leftrightarrow \text{H}_{2,e=7} + e$
3	R119 $\text{O}(^1\text{D}) + \text{H}_2 \Rightarrow \text{H} + \text{OH}$
4	R129 $\text{O}_2(\text{a}^1\Delta) + \text{O}_3 \Rightarrow \text{O}_2 + \text{O}_2 + \text{O}$
5	R3 $\text{Ar} + e \Leftrightarrow \text{Ar}^{3*} + e$
6	R60 $\text{H}_2 + e \Leftrightarrow \text{H}_{2,e=5} + e$
7	R56 $\text{H}_2 + e \Leftrightarrow \text{H}_{2,e=1} + e$
8	R11 $\text{O}(^1\text{D}) \rightsquigarrow \text{O}$
9	R16 $\text{O}_2 + e \Leftrightarrow \text{O}_{2,r=1} + e$
10	R23 $\text{O}_2 + e \Leftrightarrow \text{O}_2(\text{a}^1\Delta) + e$
11	R17 $\text{O}_2 + e \Leftrightarrow \text{O}_{2,v=1} + e$
12	R22 $\text{O}_2 + e \Leftrightarrow \text{O}_{2,v=4} + e$
13	R108 $\text{Ar}^+ + 2\text{Ar} \Rightarrow \text{Ar}_2^+ + \text{Ar}$

3.4.3.4 Conclusion

Global sensitivity analysis (GSA) of pulse discharge in 1% H₂, 0.15% O₂, and 98.85% Ar gas mixture was performed in a complicated reaction set containing 55 species and 146 reactions. The input parameters were reaction rate coefficients varied around initial values according to a prescribed log-normal distribution. The log-normal distribution was selected because it preserves the positive nature of reaction rate coefficients. For the presented simulated system, i.e., flow reactor, pulse power deposition, and H₂-O₂-Ar gas mixture, the histograms of output values indicate they follow the log-normal distribution, but with different mode and deviation of the distribution. The 99% probability intervals for all species' densities and T_e were computed from their distributions.

The output values, species' densities, and electron temperature were collected at two time intervals during the simulation: around the peak of the second pulse (time between 50.24 and 50.5 μ s) and plateau before the end of the second pulse (time between 98 and 100 μ s). Values of densities and temperatures used in the analysis were averaged values of collected values - one value for each time interval. The ground state species composing the feedstock gas, i.e.,

Ar, H₂, and O₂, showed no sensitivity to uncertainties in the reaction rate coefficients, which is most likely due to the gas inflow that completely replenishes feedstock gas species during the pulse and low ionization fraction ($\sim 10^{-4}$). Hydrogen and argon species show smaller sensitivity to input uncertainties at the peak of the pulse compared to sensitivity at the end of the pulse, as shown in Fig. 3.13. Meanwhile, oxygen species show approximately the same sensitivity in both time intervals. The results of sensitivity of electron density on the uncertainty of input parameters show that individual sensitivity index (S_i) is not always an accurate indicator of how important the reaction is, and the combined effect of reactions is required, which is provided by the total sensitivity index S_{T_i} . The most influential reaction according to sensitivity index S_i is ranked 10th according to the total sensitivity index S_{T_i} . The top nine influential reactions according to S_{T_i} (Table. 3.3) do not directly produce or consume electrons, and indirectly affect the electron density. This indicates a complicated reaction set, and assuming the reaction importance based on the participating species and reaction rate coefficients can lead to an incorrect conclusion of the reaction's importance.

3.5 Conclusion

The global sensitivity analysis (GSA) can give insight into the sensitivity of the simulation results on input uncertainties. Two different methods for conducting GSA were presented: one-at-the-time (OAT) and variance-based method. The OAT method is the simpler of the two methods and is focusing on capturing the individual effect of changes in input parameters on quantities of interest (QoI) of the simulated system. The OAT analysis would also show if the parameter has a positive or negative effect on results: will the increase in the input parameter increase or decrease the QoI? The most straightforward OAT analysis would involve only two additional values for each input parameter, one with positive and negative parameter change from the initial value, e.g., $\pm 20\%$ change. With two samples for each input parameter, the analysis would require $2p + 1$ independent model evaluations to

define whether the input parameter positively or negatively affects the QoI. Knowing the input parameter quantitative effect and mapping their positive/negative effect on results is especially important when output distributions are not symmetrical. The goal of the GSA is to assess the deviation of the quantities of interest (QoI).

In addition to the OAT, the variable-based method can give the combined effect of changes in input parameters on QoI. By employing a naïve method, acquiring the combined effect would require sampling each parameter with a given number of samples. That would result in many model evaluations, which is not practical for models beyond a few input parameters (p). Different sampling methods were developed to reduce the number of required parameters, and Saltelli sampling was used in this chapter. Saltelli presented a method of computing individual (S_i) and total sensitivity (S_{T_i}) indices to define a *relative importance* of each reaction in a reaction set to selected QoI. The individual sensitivity index defines the individual importance of reactions, and the total sensitivity index also includes the interactions between reactions.

The distribution based on simulation results computes the probability intervals of output values, e.g., densities and temperatures. The histograms of output values were consistent with the log-normal distribution for the presented simulated system, i.e., flow reactor, pulse power deposition, and presented H₂-O₂-Ar reaction set. By computing the parameters for the log-normal distribution and plotting it on top of histograms, it was confirmed that the species densities follow the log-normal distribution but with different modes and deviations of the distribution. Output probability intervals for selected species were presented and discussed. It needs to be pointed out that the output values may not have the same distribution as the input parameters. Current GSA analysis treats the reversible reactions as one reaction. It does not define the importance of the forward and backward reactions separately by computing two sensitivity indices: one set for forward and one set for backward reaction. The reverse reactions would need to be separated into two independent reactions, each with its reaction rate coefficient and uncertainties, to define the importance of each reaction.

CHAPTER 4

PRIORITIZING REACTIONS AND SPECIES

4.1 Introduction

The computational implication of including a Boltzmann equation (BE) solver into the KGMf is huge but increases the fidelity of the results. Prioritizing methods provide means to define the importance of reactions and species in a given reaction set, enabling the reduction of reaction sets reduction while preserving the target species' density. Incorporating a self-consistently evaluated EEDF from Boltzmann equation (BE) solver in prioritizing reactions and species will better capture the importance of the EEDF on the reduction process, especially when preserving the charges species density. It is expected that charged species will follow the electron density more closely and depend on the electron energy. This dependence is better described with an EEDF evaluated by a BE solver, rather than using pre-calculated values of the EEDF [89, 110] or assuming a Maxwellian EEDF [86]. When using pre-calculated EEDFs, two assumptions have to be made: the reduced electric field (E/N) range and the ratio of species densities in the gas mixture. The latter assumption is more challenging since the gas mixture changes during the reduction process as the ratio depends on species included in the reaction set. Combining the reaction set reduction with a self-consistently evaluated EEDF would offer insight into the dependence of the EEDF on the reduction process. The effect of using a self-consistently evaluated EEDF instead of a Maxwellian EEDF will be explored and results compared.

Finding dominant species and reactions involves investigating the importance of individual reactions in the reaction set, intending to classify or rank the species and reactions according to their influence on the selected species density (i.e., target species). The list of reactions ranked by importance can be used in sensitivity analysis since a smaller reaction

set enables either better sensitivity analysis or enables an analysis that would otherwise not be computationally feasible. Reaction sets can include reactions acting on different time scales, making interactions between species challenging to understand. Reactions acting on a short time scale can be linked into reaction chains, where the second reaction consumes species produced by the first reaction. By eliminating fast species, i.e., species produced and consumed on a much shorter time scale than the simulation's focus, the reaction set is reduced, and interaction between long-lived species can be observed.

Clarke [111] proposed an algorithm for determining the steady-state concentrations in a chemical system by computing the null space of the stoichiometric matrix. In that algorithm, the elements of the stoichiometric matrix were reaction rates (or reaction rate coefficients) that defined the transitions between species densities. Schilling *et al.* [112] investigated a cellular metabolic reaction network defined by enzymatic reactions and transport processes and focused on the steady-state of a given enzyme. The work concentrated on utilizing linear algebra and optimizations to find a unique set of pathways.

Lehmann [83] presented an algorithm to determine the dominant pathways in a chemical system and its application to the production/destruction of ozone in the stratosphere. The algorithm is based on linear optimization to eliminate the insignificant reactions based on species concentration and the reaction rates. The reactions with a small relative contribution to species concentration were removed from the reaction set. He modeled the species and the reaction set as a circuit, where the main objective was to find the minimal resistance of flow through the circuit. The path offering the least resistance is also the pathway that is the most dominant. In addition to finding the most dominant pathway, the algorithm can find alternative pathways and pathways that lead to the destruction of species. Lehmann [84] presented an algorithm with the following improvements: no assumption of a steady-state, pathways with rates lower than the threshold are neglected to reduce the number of reactions, and better treatment of sub-pathways. The last improvement means that if the newly generated pathway includes sub-pathways, i.e., part of the current pathway exists in

some other pathway, the pathway is split into shorter pathways, and reaction probability is computed for each sub-pathway. Reaction probabilities are computed from reaction rates.

4.2 Methods for Ranking Species and Reactions

The reaction set is defined by species, reactions between species, and reaction rate coefficients of reactions. The reaction set in low-temperature plasmas can have many species and reactions. The main idea behind reducing the reaction set is to reduce computational requirements and still capture the time evolution of selected species within a specified error range. Finding important species and reactions is beneficial as the number of species and reactions directly influences the computational complexity of simulations in two main directions: reduction of the size and reduction of the stiffness of generated ODE system. Reducing the number of species reduces the size and complexity of the ODE system, which enables simulation of a more complex reaction set on the same hardware and within the same time scale. The reduction of the number of reactions can reduce the stiffness of the ODE system and enable the use of larger time steps in the integrators, thus reducing the required time to perform the simulation [82, 87].

Reaction set reduction methods can be divided into three categories [86]: lumping, time scale analysis, and skeleton-based methods. Lumping methods are based on grouping individual species into clusters and treating species clusters as one species, effectively reducing the reaction set by reducing the number of species and reactions. Grouping species into clusters can be done based on species threshold energies, e.g., grouping all excited species from the same ground species with similar excitation energies into one excited state species. Time scale analysis methods are based on finding short-lived species and joining the reaction rates into reaction pathways. The reaction chain has a combined reaction rate computed from individual reaction rates [45, 84]. Short-lived species are treated like they are in quasi-steady-state (QSS) most of the time, and the coupled reaction rates can be replaced with an

analytical function. Skeleton-based methods are based on the idea that a skeleton (or graph) can be created for a given reaction set (Fig. 4.2). When a graphical representation of the reaction set is generated, general graph methods can be used to find non-influential species and reactions that can be removed from the graph. Skeleton-based methods can be species-oriented or reaction-oriented. The former focuses on removing species from the reaction set and then removing all reactions containing removed species. The latter focuses on ranking and removing reactions and then removing all species included in removed reactions.

From the perspective of computational complexity, species-oriented methods of reducing the reaction set are arguably better as they directly reduce the size of the ODE system. The easiest method to define the importance of individual species is to eliminate species with low densities [79]. This method is straightforward but could lead to eliminating species with low densities that are intermediate species between two high-density species and are required to describe both high-density species adequately. The principal component analysis (PCA) [113] focuses on reducing the number of variables in the simulation by computing the principal components (input variables) in the simulation. The PCA does not reduce the number of species in the reactions set but defines a reduced number of principal components that are a linear combination of densities of the original species. The simulation error minimization connectivity method (SEM-CM) [114] generates many reduced reactions sets based on normalized Jacobian, then performs the simulation and computes the error for every reduced reaction set and finds the optimum reduced reaction set. The method produces a smaller reaction set compared to other methods but requires more CPU time. The directed relation graph (DRG) method [82, 87, 88] is based on presenting the reaction set as a graph and removing the reactions with a small relative contribution to the species densities. The DRG-aided sensitivity analysis (DRGSA) extends the original DRG method by additionally removing non-important species after the reaction set is reduced by the DRG method. The DRG with error propagation (DRGEP) [115, 116] evaluates species importance based on how error caused by removing the species is propagating through the reaction chain from the

removed species to the target species. The reaction chain could have many other intermediate species between removed species and target species. The revised DRG (RDRG) method [82] is an enhanced DRG method where the skeleton mechanism is limited by the number of species instead of relative net contribution ε .

Finding the optimal reduced reaction set for selected species of interest is an interactive process that involves an initial simulation with a detailed reaction set and many simulations with reduced reaction sets. Many evaluations performed during the reduction process make volume-averaged 0D (global) models a perfect candidate for simulations. They offer a good balance between the complexity of the model and the running time of the simulations. Global models also offer great flexibility in defining the complexity level of the model used in the simulations, e.g., which surface processes or sheath approximations to include. Reduced reaction sets developed via global models that include only important species and reactions are suitable for use in more computationally intensive spatially-dependent simulations.

It is important to remember that the reduced reaction set is only valid at or around conditions that were used to perform the reduction of the reaction set, i.e., for selected target species and simulation conditions, e.g., system volume, partial pressures, temperatures, and power (energy) deposition. A reduced reaction set based on one set of simulation conditions can adequately describe quantities of interest in that regime but can completely fail to do so in a different parameter regime.

The main parameter for ranking species and reactions is a *threshold* (ε) that separates short and long-lived species in the reaction pathway method or less and more important reactions according to their relative contribution in the DRG method. There is generally no direct and linear dependence between the value of the threshold and the error in target species density, which prevents users from defining the threshold parameter according to the accepted target species density error of the reduced reaction set.

4.2.1 Reaction Pathways

Creating reaction pathways and ranking generated reaction chains according to their importance is a species-oriented group of methods that focus on finding short-lived species and removing them from the reaction set. When species are removed from the ODE system (and reaction set), they are treated as species in a quasi-steady-state (QSS), or an analytical function replaces their reaction.

Lehmann [83] first developed and later improved [84] the algorithm for finding reaction pathways and ranking them according to their contribution to the density of selected species. The reaction rates are used to compute the relative importance of individual reactions to given species' density. Short-lived species are consumed on a time scale of interest, i.e., species produced in one reaction are consumed in another reaction. Both reactions are joined into a reaction pathway (reaction chain), and a new reaction rate is computed to preserve the time evolution of other species participating in both reactions.

Based on Lehmann's algorithm presented in Ref. [84], Markosyan *et al.* [45] developed an open-source simulation code PumpKin. PumpKin creates reaction pathways by finding short-lived species consumed when two reactions are joined into a reaction pathway - some product species from one reaction are used as reactants for the following reaction in the reaction pathway. When the reaction pathway is created, the chain reaction rate is computed from the reaction rates of underlying reactions. PumpKin is distributed as source code and can be easily linked with other simulation codes or used as an external reaction pathway finder based on parameters prepared in input files. Ding and Lieberman [35, 85] used a global model in connection with PimpKin to define the dominant reaction pathways for a He/H₂O discharge.

PumpKin can only process unidirectional reactions, either forward or backward. Reverse reactions, i.e., reactions where the backward reaction rate is computed from defined forward reaction rate using detailed balancing, have to be separated into two reactions before using

them in PumpKin. Input parameters for PumpKin are a list of species and reactions (separated into forward and backward reactions), the time evolution of species' densities, the time evolution of reaction rates (not rate coefficients), a list of time values, and the stoichiometric matrix. The output from PumpKin is a list of reaction pathways (or single reactions if pathways cannot be composed) and the relative contribution of each reaction/pathway to the density of selected species.

4.2.2 Directed Relation Graph (DRG)

The directed relation graph (DRG) method presents a different approach to defining the dominant reactions. As with reaction pathways, dominant reactions are defined according to their effect on selected (target) species' density. The dominant reactions and species that compose a skeleton mechanism are always defined for selected (target) species and selected times-of-interest. The DRG method is a localized in time reaction set reduction method as the reduction is valid only for specific conditions and should not be used to develop a global skeletal mechanism that would be valid over changing conditions [87]. A global skeletal mechanism is generated by combining reduced reaction mechanisms at each time of interest. The combined reaction set will recover the target species at all times of interest but will be more complicated and detailed than individual reaction sets at each time. Combining reduced reaction sets at a large number of times could result in the detailed reaction set, i.e., the initial reaction set used as a starting point for the reduction.

The use of the DRG method in reducing reaction sets was demonstrated by Lu & Law [87] on a detailed ethylene oxidation mechanism with 70 species and 463 reactions, where the reaction set was reduced to a skeletal mechanism with 33 species and 205 elementary reactions. The reduced reaction set was validated in high-temperature chemistry in a perfectly stirred reactor (PSR) and low- to moderately high-temperature chemistry in a range of pressures and showed good agreement with a detailed reaction set. The same authors, Lu & Law [82], reduced two large reaction sets: *n*-heptane with 561 species and 2000 elementary reactions

was reduced to 188 species, and iso-octane with 857 species and 3606 reactions was reduced to 233 species. For both reaction sets, reduced reaction sets recovered the auto-ignition and PSR over wide parameter ranges. Lu & Law [88] tested the DRG method in various systems to investigate how reduced reaction sets can recover fast and slow processes. The DRG method successfully reduced the mechanism in systems with quasi-steady-state (QSS) species and partial equilibrium (PE) reactions. In systems with slow processes, it failed to capture slow processes using a single point in time, requiring multiple times to capture those slow processes.

The main parameter for the DRG method is a threshold of a relative reaction contribution to the target species' density. The threshold is a significant parameter in reducing the reaction set. However, there is no direct or linear relationship between the threshold value and the error computed from detailed and reduced reaction set simulations. Plotting the number of species and reactions depending on threshold values can give a first insight into the behavior of the reaction set for different thresholds (Fig. 4.1). Using the time evolution of

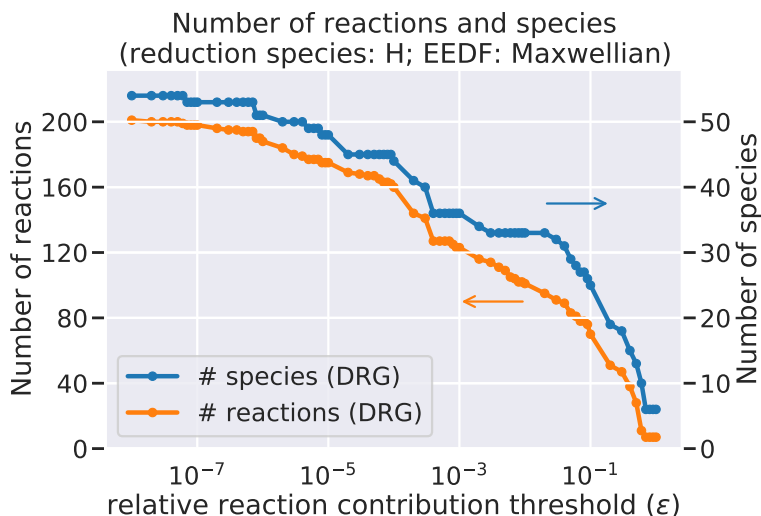


Figure 4.1 The number of species and reactions in the reduced reaction set depending on the relative reaction contribution threshold (ϵ) on the atomic hydrogen density (H). The relative contributions were computed at multiple times-of-interest (see Fig. 4.3).

reaction rates from simulation with a detailed reaction set, generating the number of species and reactions for a range of threshold values is done without running any additional simulation and usually takes a fraction of the time compared to the simulation time. Observing the plot in Fig. 4.1, there are some threshold values where the number of species changes significantly with a small change of threshold, which indicates that clusters of strongly connected species are removed from the reaction set when the threshold changes slightly. Threshold values before (or after) the significant jump (or drop) are good starting points for the initial threshold and to further evaluate the contribution of that cluster of species to the target species' density.

The threshold value has a strong effect on creating a reduced reaction set, and selecting the correct value is of great importance. The plot presented in Fig. 4.1 can serve as a good starting point for selecting the threshold, or the reduction can be made from the initially selected threshold and changing the threshold in every reduction step. The threshold in each reduction step can be defined using different plans: selecting individual thresholds based on graph in Fig. 4.1, using bisection method with defined threshold interval, or multiplying the threshold in a current step to get the threshold for the next step.

Koelman *et al.* [89] introduced an additional threshold when defining the importance of reactions to conserve better the completeness of the reaction set: electron power density. The reaction's relative contribution to the species density and the reaction's effect on the electron power density is computed for every reaction in the reaction set, and reactions having both contributions above thresholds are defined as significant. The thresholds for relative density and energy contributions have different values and are defined by the user. Although the method with added net contributions of reactions to electron power density, as presented in Koelman [89], has increased ability to capture the properties of detailed reaction set, all reaction set reductions presented in this section were made without considering the reaction's energy contributions to species densities.

4.2.2.1 Creating Graph and Skeleton Mechanism

The core component of the DRG method is a graph that presents a reaction set (Fig. 4.2) in a species-centered way. Species are assigned to nodes, and directional edges (connection with a direction between nodes) represent reactions, where the reaction rate is used as an edge weight. The direction of the edge defines the direction of the reaction from the reactant to the product species. The graph does not change during the simulation, and only edge

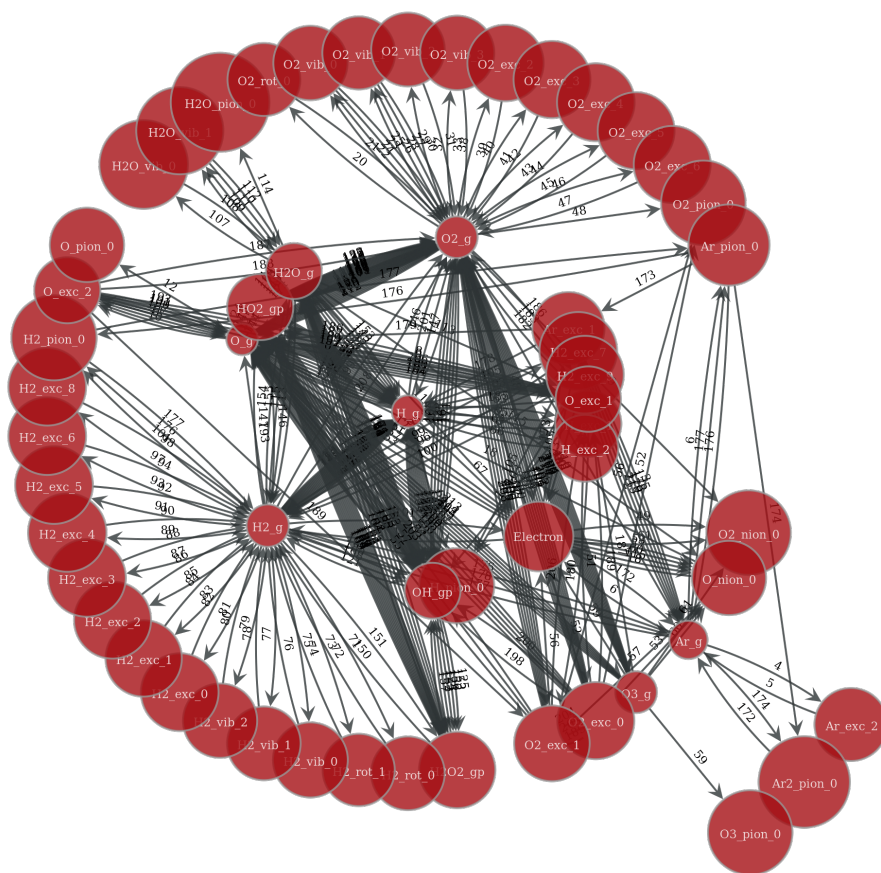


Figure 4.2 Example of a graph presenting a detailed reaction set for an $\text{H}_2\text{-O}_2\text{-Ar}$ gas mixture (55 species and 215 reactions, detailed reaction set in Table C.1). The target species, H (node labeled as "H_g"), is positioned in the center of the graph. Species in the reaction set are nodes, and reactions are edges (connections). The number on each edge is the reaction number. Forward and backward reactions are separated and have unique numbers.

weights (reaction rates) change in time. When the reverse reaction is added to the graph, two edges are added: one for the forward and one for the backward reaction, each with its reaction rate as its edge weight.

The process of creating the skeleton mechanism, i.e., the reduced reaction set, is the following:

- add all species from detailed reaction set into the graph as nodes
- add edges for all reactions in the reaction set (e.g., $A + B \rightarrow C + D$):
 - add an edge $A \rightarrow C$ from species A (reactant) to species C (product) and assign the reaction rate R_{AC} to the edge
 - add edges for all reactant (A, B) and product (C, D) species combinations, if the species A and C are different; adding edges for $A \rightarrow D$, $B \rightarrow C$, and $B \rightarrow D$
- for all nodes in the graph, compute relative net contribution r_{AC} of each reaction to the species' density; the equation for computing r_{AC} can depend on reactions that contribute and consume species in each node: some examples are presented in Eqs. (4.1)-(4.3)
- remove all edges from the graph that have $r_{AC} < \varepsilon$ (removing non-influential reactions)
- remove all unconnected nodes, i.e., nodes not connected to any other node (removing species from skeleton mechanism)

The main parameter to separate influential from non-influential reactions is the reaction's relative contribution to the density of the species in node A , and can be computed using the following equation (adapted from [88]):

$$r_j = \frac{|\nu_j R_{A,j}|}{\sum_{i=1, N_A} |\nu_i R_{A,i}|} \quad (4.1)$$

where j is the j -th reaction in node A , N_A is the number of reactions in node A , and ν_i and ν_j are the net stoichiometric coefficients of species A in reactions i and j , respectively.

When the sum of forward and backward reactions in the node is approximately the same, i.e., $|\sum_i \nu_i R_{f,i} - \sum_i \nu_i R_{b,i}| \approx 0$, the denominator in Eq. (4.1) is small, causing the relative reaction contributions to be large and not representative of the true importance of individual reaction. One way to prevent this is to separate the sum of reaction rates in the denominator into a sum of forward and a sum of backward reactions by rearranging the equation into:

$$r_j = \frac{|\nu_j R_{A,j}|}{\sum_{i=1, N_{f,A}} |\nu_i R_{f,A,i}| + \sum_{i=1, N_{b,A}} |\nu_i R_{b,A,i}|} \quad (4.2)$$

where $N_{A,f}$ and $N_{A,b}$ are the number of forward and backwards reactions in the node A , respectively. Another approach is to completely separate relative contributions from forward and backward reactions:

$$r_{f,j} = \frac{|\nu_j R_{f,A,j}|}{\sum_{i=1, N_{f,A}} |\nu_i R_{f,A,i}|} \quad r_{b,j} = \frac{|\nu_j R_{b,A,j}|}{\sum_{i=1, N_{b,A}} |\nu_i R_{b,A,i}|} \quad (4.3)$$

where subscript f indicates the forward reaction index and subscript b indicates the backward reaction.

In reaction sets without backward reactions, all proposed methods of computing r_j assign the same importance to reactions. The difference in definitions for r_j becomes evident when the reaction set includes reverse reactions (i.e., reactions with both a forward and backward reaction) with a mixture of fast reaction rates and reactions where forward and backward reaction rates are almost the same. When reaction rates of forward and backward reactions are approximately the same, the selected equation may not assign accurate importance to the individual reaction. It might instead underestimate important reactions or overestimate unimportant ones. When using net contributions given by Eqs. (4.3), one needs to take into account that the same value of net contribution threshold ε has different effect to the ranking of reactions compared to Eqs. (4.1) and (4.2).

Authors Lu & Law discussed in Ref. [88] the applicability of each equation to different types of combustion systems, where reactions are defined in Arrhenius form (Eq. (3.14)): quasi-steady-state (QSS) and partial equilibrium (PE) systems. In the QSS system, the

species are quickly consumed, and they can be treated as being in steady-state. In the PE system, some fast reverse reactions have much higher reaction rates than other reactions, and fast reactions can lead to erroneous results. These fast reverse reactions can be found in reaction sets that include externally driven reactions (e.g., laser irradiation). The authors suggest using reaction net contribution computed by Eq. (4.1) in the general case, more so, if reaction rates are defined in Arrhenius form.

Lu & Law [82] presented two-stage reaction set reduction because the relative contribution of reactions in each node changes after removing less important reactions from the reaction set. After the initial reduction of the detailed reaction set, the authors proposed to compute the new net contribution of reactions in each node and again remove the reactions with the net contribution below the given threshold. Their investigation showed that creating a skeleton mechanism is a fast operation and performing the same operation twice presents a small addition to the overall time, including time to perform the simulation using the skeleton mechanism.

4.2.3 Error Between Detailed and Reduced Reaction Sets

The last step in the reaction set reduction is to evaluate if the reduced reaction set can adequately capture the selected quantity of interest (QoI), i.e., if the error between detailed and reduced reaction set simulations are inside the prescribed maximal error δ_{\max} . The method of computing the error depends on the goal of the reduction: does the reduced reaction set need to capture the values in a steady-state, or it needs to capture the time evolution of selected QoI (e.g., species' density or temperature). For capturing the steady-state values, values from detailed and reduced reaction sets can be compared using the following equation:

$$\delta = \frac{|A_i^{\text{red}}(t_j) - A_i^{\text{det}}(t_j)|}{A_i^{\text{det}}(t_j)} \quad (4.4)$$

where index i runs over all quantities of interest (QoI), A_i is the value of QoI (e.g., density or temperature), t_j is the time at which the error is computed, superscript ^{det} and ^{red} define the system with detailed and reduced reaction set, respectively. The value of QoI, A_i , can also be the average value computed over time interval at steady state to eliminate any fluctuation of values due to numerical uncertainties in the simulation, as presented in Fig. 3.5.

When one of the goals during the reaction set reduction is to preserve the time evolution of selected QoI, the error caused by the reaction set reduction process should be computed at multiple times of interest during the simulation (Fig. 4.3). Selecting multiple times of interest for computing error is important in simulations where reaction rates and species densities fluctuate over many orders of magnitude and can profoundly affect reaction kinetics, e.g., in pulse-driven discharges. Nagy and Turányi [114] suggested the following equation to compute

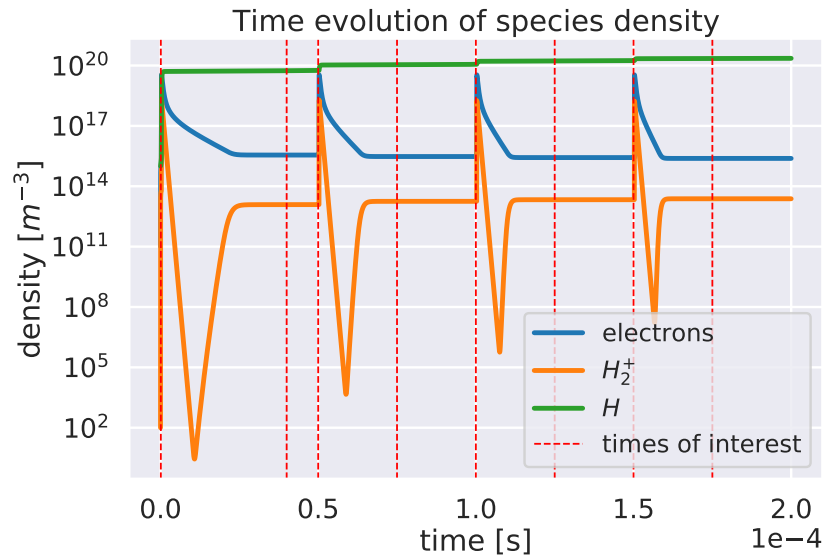


Figure 4.3 Time evolution of species densities (electrons, H, and H_2^+) and selected times of interest. The error in quantities of interest (QoI) from a detailed and reduced reaction set is computed at selected times of interest, marked with red vertical dashed lines. Multiple times of interest add to the complexity of the reduction process but decrease the error from reduced reaction set simulations in cases when selected QoI changes in time.

the error that is used to compare with the given maximum accepted error (δ_{\max}):

$$\delta = \max_{i,j} 2 \frac{|A_i^{\text{red}}(t_j) - A_i^{\text{det}}(t_j)|}{A_i^{\text{det}}(t_j) + \max_j A_i^{\text{det}}(t_j)} \quad (4.5)$$

where index j runs over the set of pre-selected times of interest t_j .

4.2.4 Reaction Set Reduction Process

The main goal of reducing the reaction set is to lower the computational requirements while still capturing the time evolution of selected species within selected acceptable error. Due to its lower computational requirement, a reduced reaction set can be used in simulations that consider the spatial dependence of the main plasma parameters. The reaction set reduction is always performed for a selected (target) species. The goal of the reduction is to preserve the density of target species in a given time interval. After successful reaction set reduction, simulations with a reduced reaction set will preserve the selected species' density. However, it might not capture the time evolution of other densities or computed parameters (e.g., ignition times) that could deviate more than maximum error, δ_{\max} , from values in detailed simulation.

Reducing the reaction set is an iterative process, where the error in results from simulations using detailed and reduced reaction sets is computed. When the error (δ), computed from Eq. (4.4) or (4.5) between two results, is larger than an acceptable error (δ_{\max}), the reduction of the reaction set is performed again with a smaller reaction rate threshold ε . The reduction process is terminated when one of the following conditions is reached: the error is acceptable, the reaction set is not changing, or the maximum number of attempts is reached. When the error between reduced and detailed reaction set, δ , is below a user-defined acceptable error δ_{\max} , the reduction process is stopped. The reduction process is also stopped when reaction sets from two consecutive reduction steps have the same species and reactions. The further reduction will most likely generate the same reaction sets, and a further reduction will not be beneficial. The last possible condition for terminating the reduction process is

when the number of reductions reached the maximum number of attempts. It serves as a last resource in preventing the indefinite reduction process. The procedure of reducing the reaction set using the KGMf and the DRG method is as follows (Fig. 4.4):

1. the KGMf simulation with detailed reaction set is run and data needed for reaction set reduction is saved into files: time evolution of species' densities, the time evolution of reaction rates, stoichiometric matrix, and lists of species and reactions
2. initial relative reaction threshold, ε , and the maximum allowed error, δ_{\max} , are defined
3. reduction of the reaction set for current threshold is performed; additional checks of the final reaction set are performed to make sure all necessary species are included (e.g., species defined as inflow and outflow species)
4. simulation with reduced reaction set is run
5. error, δ , between detailed and reduced reaction set simulations is computed
6. if $\delta > \delta_{\max}$, a new threshold ε is selected, and the next loop of reduction is made starting with step 3; otherwise, the reduced reaction set is accepted

The process of finding the minimal reduced reaction set recovering target species' density can be described by the main reduction parameters in each reduction step: reaction net contribution threshold ε , the number of species and reactions, and error between detailed and reduced reaction set. Table 4.1 shows an example of reaction set reduction steps, and other reaction set parameters.

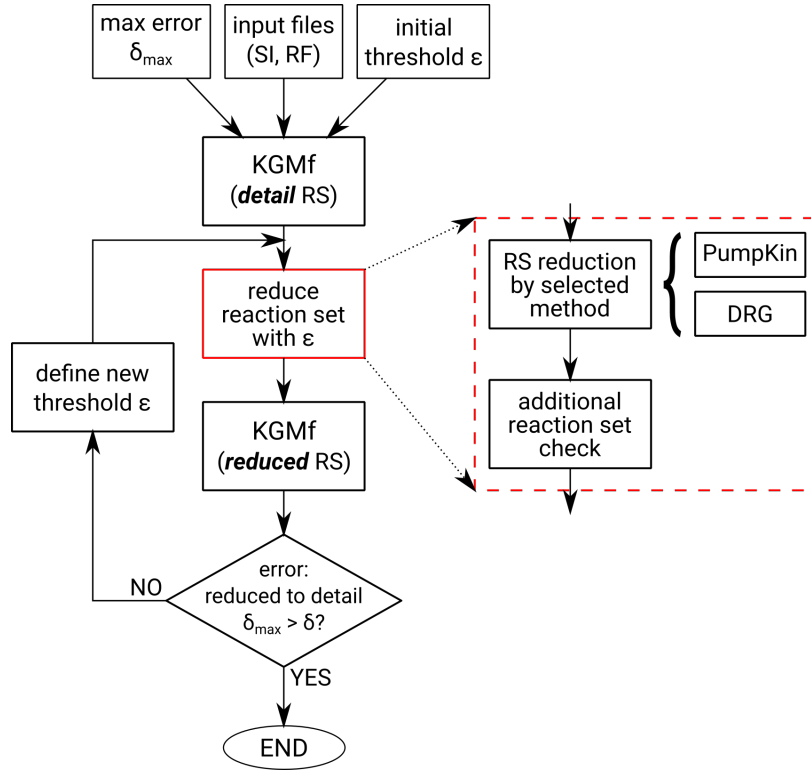


Figure 4.4 A flowchart of reaction set reduction with the KGMf. In the top row are input parameters for reaction set reduction, and the center part shows the loop required to reduce the reaction set until the given maximum error threshold is reached. *Note:* 'RS' stands for 'reaction set'.

4.3 Plasma Discharge in H₂-O₂-Ar Gas Mixture

The reaction set reduction presented in this section was performed in simulations with H₂-O₂-Ar gas mixtures and with two different target species (species of interest): ground atomic hydrogen (H) and molecular hydrogen ions (H₂⁺). Densities of selected species show different time evolutions in Fig. 4.3 regarding applied pulses. The ground atomic hydrogen (H) is reactive species, participating in 30 reactions in the reaction set and only slightly responds to applied pulses, showing a steady monotonic increase in time with small jumps at the beginning of each pulse. In contrast, the molecular hydrogen ions (H₂⁺) show large fluctuations due to applied pulses, and preserving its density would present a more challenging task. Reaction set reduction was performed using the same system (geometry and simulation pa-

rameters), and the same reaction set was used in the global sensitivity analysis in Chapter 3.

The main simulation parameters were:

- rectangular flow reactor with dimensions ($L \times W \times H$): $6.5 \times 1.5 \times 0.5$ cm ($V = 5$ cm³)
- gas mixture of 0.15%O₂ - 1%H₂ - Ar, constant pressure $p = 300$ Torr, and constant gas temperature $T_g = 500$ K
- electron temperature T_e was computed depending on used EEDF: when Maxwellian EEDF was assumed, the T_e was computed using the electron energy balance equation, when EEDF was defined by the BE solver, the T_e was computed from EEDF
- sheath losses and losses to the wall were neglected

The main reaction set reduction parameters were:

- initial reaction net contribution threshold $\varepsilon_0 = 0.1$ (10%)
- reaction threshold in each reduction step was computed as: $\varepsilon_i = 10^{-i}$, where i is the reduction step (1, 2, ..., N)
- the accepted error computed by DRG method, δ_{\max} , was 10^{-4}
- the reduction process was completed when the error from DRG method was below δ_{\max} or when two consecutive reductions resulted in reaction set containing the same species and the same reactions

The detailed reaction set was composed of 55 species (Tab. 3.2 in Section 3.4.3) and 146 reactions (Tab. C.1 in Section C). For reaction set reduction, reversible reactions were split into forward and backward reactions, treated separately by the DRG method. With forward and backward reactions treated separately, the total number of reactions was 215.

4.3.1 Capturing H density

The first reaction set reduction was made to preserve the time evolution of atomic hydrogen (H) density. Before running the reduction steps involving the KGMf simulations, the reaction set analysis with the DRG method was run using the time evolution of reaction rates and densities from the detailed reaction set simulation. The purpose of the analysis is to compute the dependence of the reduced reaction sets size (the number of species and reactions) on the reaction contribution threshold. The analysis is run without running the underlying KGMf simulations in each reduction step; just the DRG method is run to compute the reduced reaction set. The analysis will not give any information of expected error in species densities in each reduction step. Large jumps in the number of species can indicate the existence of clusters of species removed from the reaction set at the same time at specific threshold values. Further comparison of reaction sets is necessary to determine which species were removed from the reaction set at each threshold value. The plot in Fig. 4.5 can be used to define the initial value of the reaction threshold ε to reduce the required time to perform the reaction set reduction. For the presented reaction set, the analysis for one species and eight (8) times-of-interest took around 12 seconds using an average desktop computer.

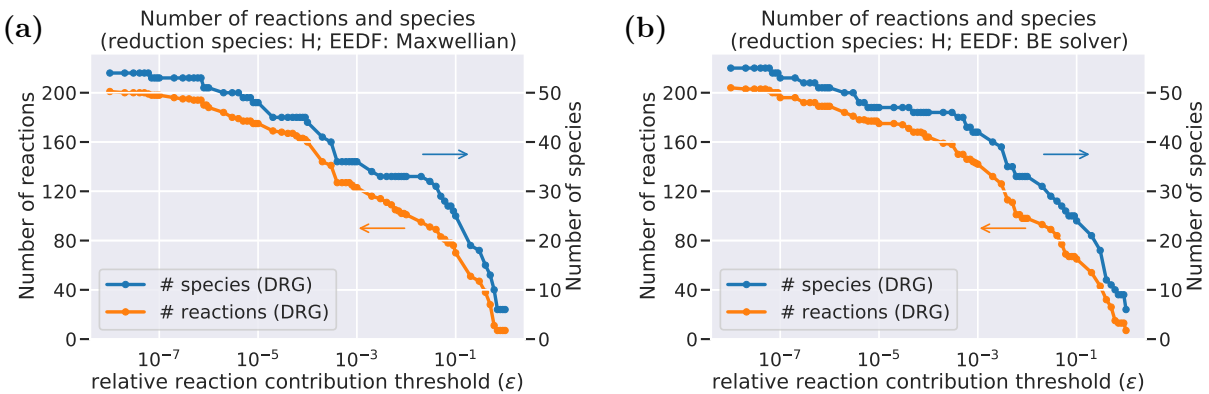


Figure 4.5 The number of species and reactions vs. relative net contribution of reactions (ε) on H density with Maxwellian EEDF (a) and EEDF from coupled Boltzmann equation solver (b). *Note:* For reaction set reductions, forward and backward reactions are counted separately, making the total number of reactions 215, while the KGMf reaction set has 146 reactions.

Table 4.1 Reaction rate threshold, number of species and reactions, and error in H density from DRG method in each reduction step for two types of EEDFs: Maxwellian EEDF (top) and EEDF from BS solver (bottom). The error was computed between simulation results from detailed and reduced reaction set for the time intervals shown in Fig. 4.3. The last column presents simulation time reduction in the KGMf with a given reaction set compared to simulations with a detailed reaction set. **Note:* For reaction set reductions, forward and backward reactions were treated separately, thus making the total number of reactions 215, the KGMf reaction set has 146 reactions.

EEDF: Maxwellian					
Step #	Reaction net contribution threshold	# species	# reactions*	Error	Simulation time
1	10^{-1}	25	70	–	(failed to converge)
2	10^{-2}	33	101	$4.49 \cdot 10^{-2}$	119 s (-68.8 %)
3	10^{-3}	36	123	$1.59 \cdot 10^{-2}$	138 s (-63.8 %)
4	10^{-4}	44	160	$1.73 \cdot 10^{-3}$	177 s (-53.7 %)
5	10^{-5}	48	175	–	(failed to converge)
6	10^{-6}	51	188	$9.17 \cdot 10^{-5}$	285 s (-25.3 %)
detailed		55	215	–	382 s (100 %)

EEDF: from the BE solver					
Step #	Reaction net contribution threshold	# species	# reactions*	Error	Simulation time
1	10^{-1}	24	65	–	(failed to converge)
2	10^{-2}	33	98	$5.20 \cdot 10^{-2}$	1447 s (-27.6 %)
3	10^{-3}	42	142	$8.10 \cdot 10^{-2}$	1735 s (-13.2 %)
4	10^{-4}	46	164	$3.54 \cdot 10^{-2}$	1632 s (-18.3 %)
5	10^{-5}	47	175	$5.74 \cdot 10^{-3}$	1762 s (-11.8 %)
6	10^{-6}	51	189	$3.84 \cdot 10^{-2}$	1925 s (-3.7 %)
7	10^{-7}	53	196	$1.72 \cdot 10^{-2}$	1839 s (-8.0 %)
8	10^{-8}	55	204	$1.56 \cdot 10^{-2}$	1873 s (-6.3 %)
9	10^{-9}	55	208	$1.40 \cdot 10^{-2}$	2063 s (3.2 %)
10	10^{-10}	55	209	–	(failed to converge)
11	10^{-11}	55	210	$4.76 \cdot 10^{-3}$	2025 s (1.3 %)
detailed		55	215	–	1999 s (100.0 %)

The reaction set reduction is an iterative process, where species and reactions are removed from the reaction set in each interaction. The simulation is performed with a reduced reaction set in each iteration. The reduction was made using the maximum acceptable error $\varepsilon = 10^{-4}$ of H density, and Table 4.1 shows reaction rate threshold and error between results from detailed and reduced simulations in each reduction step. The reduction process in both

cases (with Maxwellian EEDF and EEDF from the BE solver) was stopped when reaction sets in consecutive reduction steps yield were the same. The lower computational intensity of simulations with Maxwellian EEDF is noticeable when comparing runtimes with simulation with EEDF from the BE solver. When reducing the reaction set with assumed Maxwellian EEDF, the simulations show quicker convergence in error and the number of species and reactions, requiring only six (6) reduction steps to reach the minimum reaction set. For H density error below 2 %, only three (3) reduction steps are required. On the other hand, the simulation with an EEDF from the BE solver showed slow convergence of reduction step, requiring eleven (11) steps to reach the minimum reaction set. On top of much longer simulation steps, the error from reduced reaction sets converges slower than the Maxwellian EEDF case and requires five (5) steps to reach an error below 2 %.

Figure 4.6 shown the time evolution of H density from simulations using different EEDFs, showing a good match between evolutions and suggesting that selection of EEDF does not play a huge role in the evolution of H density. The maximum difference between densities is up 32 %, which could be sufficient for some applications.

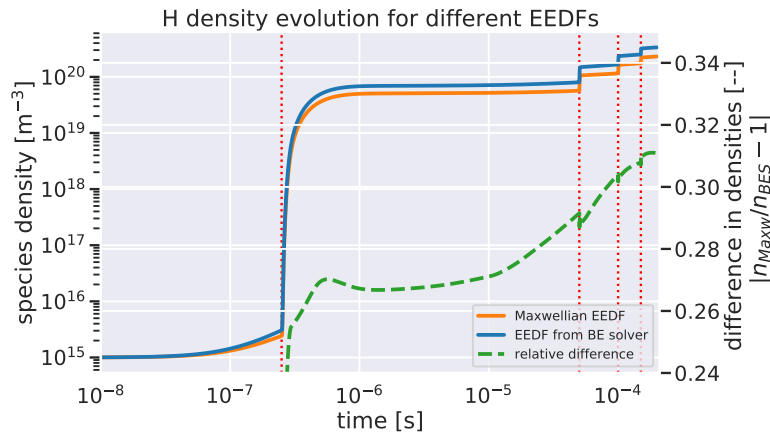


Figure 4.6 Time evolution of H density from simulations with Maxwellian EEDF and EEDF from the BE solver. The assumption of Maxwellian EEDF causes up to 32 % error in H density, which could be acceptable for some applications. Vertical red dashed lines mark the start of each pulse.

Figure 4.7 shows the time evolution of atomic hydrogen density from detailed and reduced reaction sets for both types of EEDF. The H density changes at the start of every pulse, following a considerable spike in electron density, but show a constant value between pulses and otherwise does not follow the electron density evolution and indicates that electron density does not directly influence H density. This indicates that the H density is indirectly dependent on electron density. Simulation results from all reduced reduction steps captured the time evolution of H density, except in reduction step 1, where the simulation failed to converge due to an incomplete reaction set.

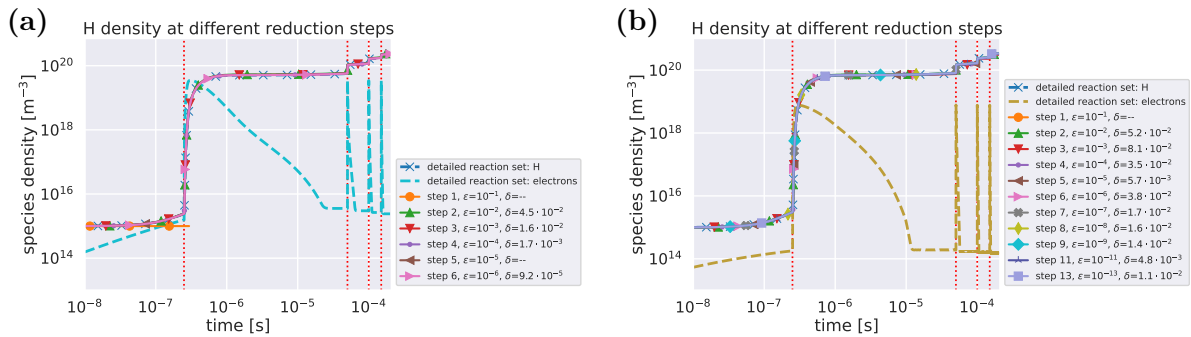


Figure 4.7 Time evolution of H density from detailed and reduced reaction sets using Maxwellian EEDF (a) and EEDF from the BE solver (b). Vertical red dashed lines mark the start of each pulse.

Figure 4.8 presents the relative errors between density from detailed and reduced reaction sets. The plot for Maxwellian EEDF (Fig. 4.8(a)) shows a clear correlation between reaction relative contribution threshold ε and relative error in H densities: a smaller threshold value results in a smaller error introduced by the reduced reaction set. On the contrary, Fig. 4.8(b) for an EEDF from the BE solver, this correlation is not noticeable, especially in the time after the first pulse, when the density increases the most. The error computed by the DRG method, also included in Table 4.1 in column 5, is displayed in the figure legend for each reduction step. For reduction with Maxwellian EEDF, the error computed by the DRG method is very similar to the relative error in H densities after the first pulse. Good correlation between error computed by DRG method and relative error from simulations

additionally simplifies the reduction process because the maximum acceptable error δ_{\max} in the DRG method can be set to the value of acceptable density error from a reduced reaction set. The reductions with EEDF from the BE solver show a different behavior, where error from densities in reduced steps does not follow assigned relative reaction contribution. At the end of the fourth pulse ($t = 2.0 \cdot 10^{-4}$ s), errors are all grouped in the error interval from $5.7 \cdot 10^{-3}$ to $8.1 \cdot 10^{-2}$. This indicates the limits for the accuracy of H density from reduced reaction sets and suggests that reduction steps beyond step 5 have only a small added value.

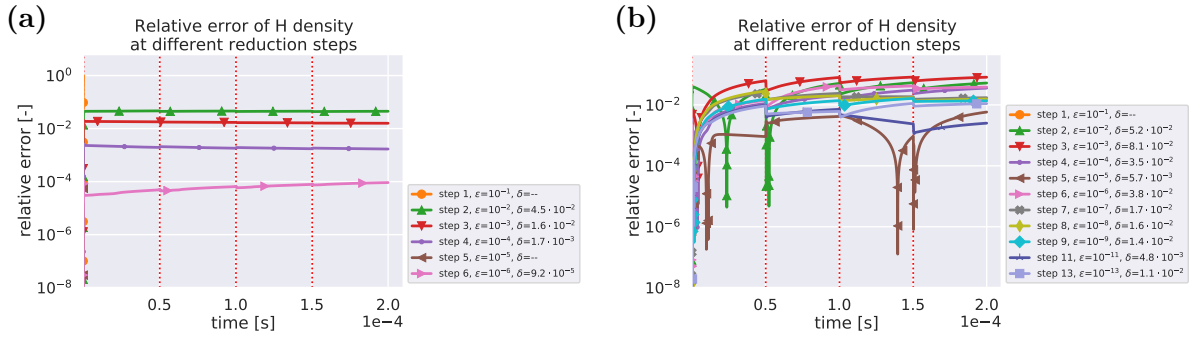


Figure 4.8 Time evolution of relative error in H densities introduced by reduced reaction sets with Maxwellian EEDF (a) and EEDF from BE solver (b). Relative errors introduced by reduced reaction sets are smaller with each additional reduction run. Hydrogen atom density at times-of-interest (Fig. 4.3) were used to compute errors in the DRG method (δ). The error in the DRG method is added to the legend for each reduction step and is very close to the relative error in H species densities in the left figure.

4.3.2 Capturing H_2^+ density

The second example of reaction set reduction was focusing on capturing the time evolution of charged species density (H_2^+), which shows large fluctuations throughout each pulse cycle (Fig. 4.3). Figure 4.9 shows the results from reaction set analysis, plotting the number of species and reactions in the reduced reaction set at each threshold value. The dependence of the number of species and reactions is similar to the one calculated for H density (Fig. 4.5). Detailed investigation shows that the number of species and reactions are the same for both species (H and H_2^+) for reaction threshold values $\epsilon \leq 3 \cdot 10^{-2}$.

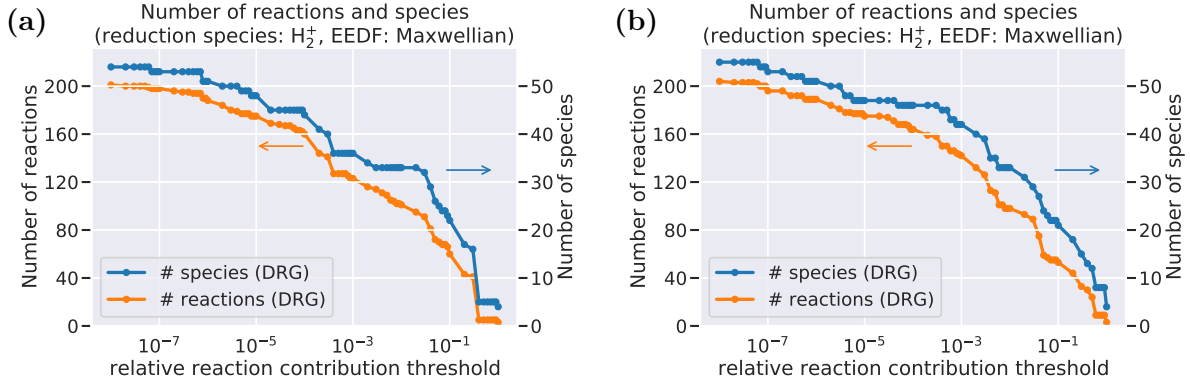


Figure 4.9 Dependence of the number of species and reactions on the relative net contribution of reactions on H_2^+ density using Maxwellian EEDF (a) and EEDF from the BE solver (b). *Note:* For reaction set reductions, forward and backward reactions are counted separately, making the total number of reactions 215, while the KGMf reaction set has 146 reactions.

The process of finding reduced reaction sets and their relative errors in time evolution of H_2^+ density started with the initial reaction contribution threshold $\varepsilon = 0.1$ and acceptable error $\delta_{\max} = 10^{-4}$. The threshold for each reduction set was computed as 10% of the threshold from a previous step. The reduction process for reductions with the Maxwellian EEDF concluded with the sixth step because the following reaction set reduction generated the same reaction set as the reaction set in the last reduction step. For the same reason, reduction with EEDF from the BE solver concluded in the eleventh step. Individual reduction steps, including the number of species and reactions, error from the DRG method, and the KGMf simulation time with each reduced reaction set, are gathered in Tab. 4.2. Two significant differences can be observed from the results presented in Tab. 4.2: the Maxwellian EEDF reduction shows a more significant reduction in time and lower errors from reduced reaction set compared to reductions with EEDF from BE solver.

Table 4.2 Reaction rate threshold, number of species and reactions, and error in H_2^+ density from the DRG method in each reduction step for reductions with Maxwellian EEDF and EEDF from the BE solver. The error was computed between simulation results from detailed and reduced reaction sets in time intervals shown in Fig. 4.3. The last column presents simulation time reduction in the KGMf with a given reaction set compared to simulations with a detailed reaction set. *Note: For reaction set reductions, forward and backward reactions were treated separately, thus making the total number of reactions 215, while the KGMf reaction set has 146 reactions.

EEDF: Maxwellian					
Step #	Reaction net contribution threshold	# species	# reactions*	Error	Simulation time
1	10^{-1}	22	60	–	(failed to converge)
2	10^{-2}	33	101	$3.86 \cdot 10^{-2}$	120 s (-68.7 %)
3	10^{-3}	36	123	$1.53 \cdot 10^{-2}$	138 s (-63.9 %)
4	10^{-4}	44	160	$1.69 \cdot 10^{-3}$	179 s (-53.2 %)
5	10^{-5}	48	175	–	(failed to converge)
6	10^{-6}	51	188	$7.65 \cdot 10^{-5}$	288 s (-24.7 %)
detailed		55	215	–	382 s (100.0 %)

EEDF: from the BE solver					
Step #	Reaction net contribution threshold	# species	# reactions*	Error	Simulation time
1	10^{-1}	21	53	–	(failed to converge)
2	10^{-2}	33	98	$9.01 \cdot 10^{-2}$	1441 s (-27.9 %)
3	10^{-3}	42	142	$6.14 \cdot 10^{-2}$	1725 s (-13.7 %)
4	10^{-4}	46	164	$1.20 \cdot 10^{-2}$	1631 s (-18.4 %)
5	10^{-5}	47	175	$1.08 \cdot 10^{-2}$	1767 s (-11.6 %)
6	10^{-6}	51	189	$1.34 \cdot 10^{-2}$	1943 s (-2.8 %)
7	10^{-5}	53	196	$3.91 \cdot 10^{-3}$	1831 s (-8.4 %)
8	10^{-4}	55	204	$6.24 \cdot 10^{-3}$	1887 s (-5.6 %)
9	10^{-5}	55	208	$7.05 \cdot 10^{-3}$	2061 s (3.1 %)
10	10^{-6}	55	209	–	(failed to converge)
11	10^{-5}	55	210	$8.57 \cdot 10^{-3}$	2020 s (1.1 %)
detailed		55	215	–	1999 s (100.0 %)

Figure 4.10 shown the time evolution of H_2^+ density from simulations using different EEDFs, showing higher values of H_2^+ density from simulation with Maxwellian EEDF in each pulse as well as in time between pulses. This confirms that Maxwellian EEDF overestimates the high energy electrons that give rise to electron impact reactions, directly impacting H_2^+ density. Maxwellian EEDF causes H_2^+ density to respond more to pulses, rising higher and dropping lower after the pulse and having a higher plateau between pulses.

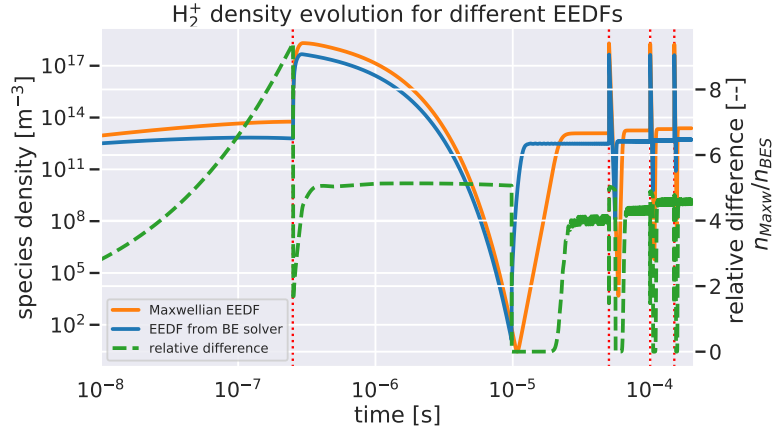


Figure 4.10 Time evolution of H_2^+ density from simulations with Maxwellian EEDF and EEDF from the BE solver. The assumption of Maxwellian EEDF causes up to 8 times higher densities (green dashed line). Vertical red dashed lines mark the start of each pulse.

Figure 4.11(a) shows good agreement of time evolution of H_2^+ density from simulations with Maxwellian EEDF with detailed and reduced reaction sets for reduction steps 3, 4, and 6. Reaction set in the reduction steps one and six could not capture the H_2^+ density, and the simulation failed to converge. The simulation in the reduction step 2 captured density fluctuations on the rising side of the pulse but failed to completely capture density drop after the pulse, causing the H_2^+ density to start raising sooner and kept it high between pulses.

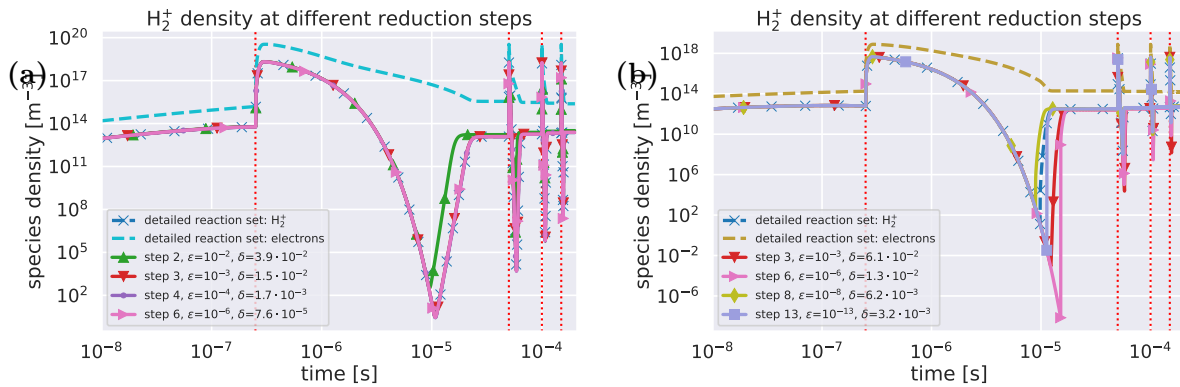


Figure 4.11 Time evolution of H_2^+ density from detailed and reduced reaction sets using Maxwellian EEDF (a) and EEDF from the BE solver (b). The H_2^+ density follows the electron density during a single pulse. For readability reasons, only selected reduction steps are shown. Vertical red dashed lines mark the start of each pulse.

The H_2^+ density rises with rising Ar^+ density due to the reaction R110 ($\text{Ar}^+ + \text{H}_2 \Rightarrow \text{Ar} + \text{H}_2^+$). The Ar^+ density rises sooner in step 2 compared to the other steps due to the EEDF having a higher energy tail causing a higher probability of ionization of argon (reaction R4: $\text{Ar} + e \Rightarrow \text{Ar}^+ + 2e$). The time evolution of H_2^+ density from reduction step 2 is closer to density from detailed reaction set with EEDF from the BE solver, raising quicker after the pulse. Figure 4.11(b) shows the time evolution of H_2^+ density from simulations with EEDF from the BE solver with detailed and selected reduced reaction sets. The densities agree well when the electron density is constant and shows huge differences when electron density changes. The H_2^+ density from reduced reaction sets cannot correctly recover the density raise after the pulse, underestimating the density for a factor up to 10^8 (reduction step 6).

The error in the reduction step 2 of the reduction with Maxwellian EEDF is even more profound in Fig. 4.12(a), where relative errors of H_2^+ density for all reduction steps are plotted. A huge error for density from reduction step 2 is misleading in Fig. 4.12(a), as it indicates that the amplitude of density is different, but in reality, the cause of this considerable

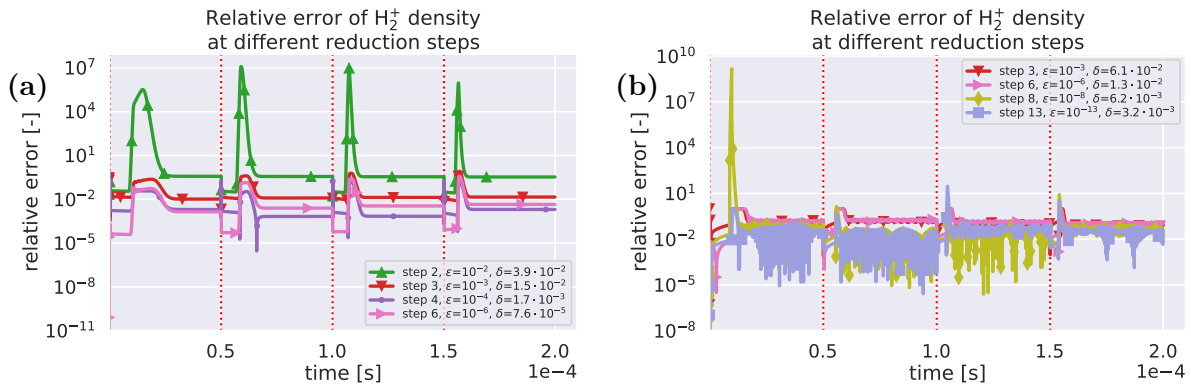


Figure 4.12 Time evolution of relative error in H_2^+ densities introduced by reduced reaction sets with Maxwellian EEDF (a) and EEDF from the BE solver (b). The H_2^+ density follows the electron density during a single pulse. Errors computed in the DRG method (δ) are displayed in the legend for each reduction step. The plot of relative errors (b) shows that errors introduced by reduced reaction sets could be orders of magnitude different from the supplied reaction threshold ϵ , meaning there is no direct connection between the two values. For readability reasons, only selected reduction steps are shown. Vertical red dashed lines mark the start of each pulse.

error is a time shift of the reduced reaction set density. Relative errors from reduction steps 3, 4, and 6 show a tendency to be smaller for smaller reaction contribution threshold ε , but are all around 10^{-2} between pulses when electron density is constant. Figure 4.12(b) shows errors in the reduction steps with EEDF from BE solver, showing huge errors after the first pulse ($t \sim 10^{-5}$ s) and maximum errors around 10^{-2} for times when electron density is not changing. Error plots in Fig. 4.12 show there is no direct correlation between the reaction's relative contribution and error in species density, making it difficult to specify the relative contribution of reaction to reduce the reaction sets.

4.3.3 Conclusion

The species and reaction importance was investigated in pulse plasma discharges in an H₂-O₂-Ar gas mixture for two different species, atomic hydrogen (H) and molecular hydrogen ions (H₂⁺), and two different electron energy distribution functions (EEDFs), Maxwellian EEDF and EEDF from Boltzman equation (BE) solver. The main difference between observed species densities is their time evolution in the pulsed discharge. The H density is monotonically rising in time (Fig. 4.7) with small stairs-like jumps at the beginning of each pulse due to the increase in density of excited argon (Ar^{*}). The most important reactions for atomic hydrogen (H) density at pulse peak, determined with a global sensitivity analysis (not shown here), are argon excitation reaction R2 (Ar + e \Leftrightarrow Ar^{*} + e) and conversion of molecular hydrogen into two atomic hydrogen due to collision with excited argon defined with reaction R112 (Ar^{*} + H₂ \Rightarrow Ar + H + H). On the other hand, the H₂⁺ density (Fig. 4.11) follows the electron density changes more closely and changes many orders of magnitude during a single pulse. Different natures of density changes in time can also be observed in errors from reduced reaction sets. A relative error in H density is strongly linked to the supplied maximum acceptable error δ_{\max} and reaction net contribution threshold ε . On the other hand, a relative error in H₂⁺ density shows no connection to both supplied error values (δ_{\max} and ε). Relative density errors in reductions with Maxwellian EEDF and

EEDF from the BE solver are connected to electron density changes. If the electron density is constant, errors stay constant for both species (H and H_2^+). Changing electron density is causing raising errors in densities from reduced reaction sets, but without huge difference between species ((H and H_2^+) or EEDF used in simulations (Maxwellian EEDF and EEDF from BE solver).

The main reason for huge relative errors in H_2^+ density evolution in reduction with Maxwellian EEDF, especially in reduction step 2, is the time shift of the density and not the amplitude of the density. As presented in Fig. 4.11(a), the density from the 2nd reduction step does not capture the density drop after the pulse and starts rising sooner than the corresponding density in the detailed reaction set. That creates a huge error between densities (at the time around $\sim 0.75 \cdot 10^{-4}$ s), which appears in Fig. 4.12(a) as a huge spike in relative error. The early rise of the density after the pulse also results in higher density between pulses. Reaction sets from consecutive reduction steps (i.e., steps number 3, 4, and 6) can capture H_2^+ density in the entire pulse, and relative errors are all around 10^{-2} . Similar results can be observed in reduction with EEDF from the BE solver. The biggest error is recorded in reduction step 8 (Fig. 4.12(b)) and is caused by the time shift of the rise of the H_2^+ density and not by the higher value of the density (Fig. 4.11(b)). Errors from other reduction steps are much smaller and are all around 10^{-2} in times between pulses.

In all reaction set reductions, the reduction process finished before errors from the reduced reaction set simulations would be below given acceptable error $\delta_{\text{max}} = 10^{-4}$. Further reaction set reduction could reduce the error even more but would increase the size of the reaction set, eventually resulting in the detailed reaction set. That would defeat the purpose of reducing the reaction set in the first place.

4.4 Conclusion

Prioritizing species and reactions was presented to reduce the reaction set while preserving the time evolution of selected species densities. Two different EEDFs were considered: Maxwellian EEDF with temperature obtained from an energy equation and an EEDF from a coupled Boltzmann equation (BE) solver. To better capture the behavior of species densities during the simulation due to the pulse-driven discharge, the error in densities between detailed and reduced reaction sets was computed at different times during the simulation, at the pulse peaks and the plateau at the end of the pulse (Fig. 4.3).

From the presented results, we conclude that the user-specified reaction threshold ε is in most cases not correlated to a relative error in densities between detailed and reduced reaction sets. The maximum relative errors in densities between pulses have a constant value around 10^{-2} but show huge spikes during pulses. The spikes in relative errors of H_2^+ density (Fig. 4.12) at pulse edges are related to the time shift of density evolution and not to the amplitude of density, indicating the reduced reaction sets cannot capture the time evolution of density. For species with highly fluctuating density, regardless of the EEDF used, when the density changes over many orders of magnitude, the user-supplied reaction threshold ε still affects the simulation error: a smaller acceptable error will result in a smaller relative error from a simulation. As can be observed in Fig. 4.12(b), it is not possible to predict the relative error from a simulation based on the supplied relative reaction contribution threshold ε and the maximum error from a simulation can be several orders of magnitude different from ε .

The presented results in pulsed plasma discharges in $\text{H}_2\text{-O}_2\text{-Ar}$ gas mixture suggest that the best approach to the reaction set reduction is gradually reducing the relative reaction contribution threshold until the computed error is below the acceptable value. The reduction process requires many simulation runs, one run for each reduced reaction set. Depending on the run time of the simulation, even when the reduction process is done in connection with a

global model, the total required time for reaction set reduction can be long but still shorter compared to the fluid or PIC simulations with the detailed reaction sets.

CHAPTER 5

CONCLUDING REMARKS AND FUTURE WORK

Before extending the KGMf with the latest features, the KGMf offered many features for modeling systems that could be represented with global models - from gas-phase combustion systems to low and high-pressure plasma discharges with various predefined electron energy distribution functions (EEDF). One main focus of the research presented in this dissertation was to extend the functionality of the KGMf by adding the ability to self-consistently evaluate the electron energy distribution function (EEDF) from the system state and to rank the importance of species and reactions in a given reaction set. The former serves to increase the fidelity of the results from the KGMf. The latter was needed for the KGMf to conduct the sensitivity analysis and reaction set reduction. A great deal of effort was put into making the KGMf more user-friendly, either making warning or error messages more descriptive or simplifying the user interaction with the KGMf. For users that want to customize the KGMf to fit their needs, the testing subsystem was added to the KGMf to ensure known solutions can be reproduced.

The ability to self-consistently evaluate the EEDF was implemented by coupling the KGMf with two existing Boltzmann equation (BE) solvers. The integration of BE solvers was designed to enable future extensions and the addition of new solvers. The flexibility of implementation was demonstrated by coupling the KGMf with two different solvers; they are different in their implementation languages (Python vs. MatLab) and underlying mathematical representation of the EEDF (two- versus up to 10-term approximation). The implementation of BE solvers was verified with two different tests: benchmark with ZDPlasKin and application in high power microwave (HPM) breakdown. The benchmark of the KGMf with ZDPlasKin, both coupled with a different BE solver using a two-term spherical approximation of the Boltzmann equation, included comparing the temporal evolution

of species densities, reaction rates, and EEDFs. The steady-state densities were compared in two cases: first, for a given absorbed power density with a range of pressures; and second, for a given pressure with a range of absorbed power densities. In both cases, the steady-state densities of electrons, excited species, and ions from the KGMf and ZDPlasKin show good agreement. In applying the global model to high power microwave (HPM) breakdown, the results from the KGMf and ZDPlasKin were compared to results from PIC simulations. The breakdown times from the KGMf and ZDPlasKin with self-consistently computed EEDFs show good agreement with the breakdown time trend from PIC simulation results. The KGMf with coupled BE solver can increase the fidelity of the results and shows a relatively significant impact on the total simulation run time.

The second group of added functionalities was uncertainty quantification and reaction set reduction, both dealing with the importance of individual species, reactions, and chains of reactions in a given reaction set. Sensitivity analysis (SA) was added to the KGMf by building on the existing functionality of the KGMf. SA parameters are defined in a separate input file. The results from individual cases are stored in the same format as in a standalone simulation run, enabling reusing all existing post-processing tools. Parameters selection and value definition for SA runs are all made in the SA input file as changing single parameters, groups of parameters, or defining the parameters' values according to selected distribution. The same goes for collecting results from all individual simulations: quantities of interest (QoI) and time intervals for result collecting are all defined in the input file. Collected results from SA runs are stored in a single file. The reaction set reduction is aimed to allow users to reduce the given reaction set for a given target variable (species density or temperature). Using either PumpKin as an external utility or an internal directed relations graph (DRG) method, the KGMf can read and process reaction sets and generate reduced reaction sets according to given relative reaction contribution thresholds. Both SA and DRG methods were implemented with great care to avoid any obvious bugs or errors, but additional verification is required.

The full set of added models enables high fidelity self-consistent calculation of transient plasma evolution with time-changing EEDF for complicated collisionality in multi-component gas mixtures. Ranking of species and reaction chains can be used as input to fluid and PIC codes to provide a tractable reduces species and reaction set that maintains a user-definable level of fidelity with reduced computational cost. The minimal cost of the KGMf allows highly parameterized studies across parameter ranges which can be used as the course component for mapping behavior in conjunction with a small number of refined fluid or PIC simulations.

5.1 Future Work

There are many possible avenues for future work that will extend the application of the KGMf either by increasing the fidelity of the results or by addressing the shortcomings of the current models. Some possible avenues of future work are presented below.

5.1.1 The KGMf and EEDF Solvers

The KGMf was coupled with Boltzmann equation (BE) solvers to improve the fidelity of the results, which came at the cost of a slower simulation speed. The computational increase when simulating with coupled BE solver is not negligible, and we implemented an evaluation frequency to mitigate the effect of the coupled BE solver on overall run time. The fidelity of results would improve by adding the possibility to evaluate EEDFs based on multiple concurrent threshold parameters (multiple species density, temperature), either fulfilling one of the defined threshold values ('or' condition) or all of them ('and' condition). Possible improvements regarding coupled BE solvers could be put in two groups: optimizing the evaluation frequency of the BE solver and coupling with a faster BE solver. The two groups of improvements are independent and can independently contribute to the faster overall simulation speed of the KGMf.

One possible short-term improvement would be adding the possibility to read pre-calculated EEDFs based on selected plasma parameters, e.g., EEDF evaluated by BOLSIG+ or BOLOS for a list of reduced electric fields or electron temperatures. The KGMf would interpolate EEDFs between pre-calculated points when needed. This would allow users to transition to the KGMf by reusing already generated EEDFs and building confidence that KGMf can be used for their research. The pre-calculated EEDFs could be read from a supplied file or by connecting to a local or remote database.

In terms of coupling the KGMf with a faster BE solver, especially if one limits themselves to the two-term BE solvers, three possible choices are:

- BOLSIG+ [96]: probably the most known and used two-term BE solver. One of the great properties of the solver is grid caching - to speed up the evaluation of the EEDF, a previously computed EEDF is returned in cases when the E/N value is 'close' to what was previously used. The exact speedup of the KGMf+BOLSIG+ is difficult to predict, especially when BOLSIG+ is written in Fortran, and coupling with Python would add some overhead.
- MCIG: Monte Carlo based Boltzmann equation solver to provide high-order EEDF evaluations
- LoKI-B (LisOn KInetics Boltzmann) [66]: is an open-source simulation tool for non-magnetized non-equilibrium low-temperature plasmas excited with DC or high-frequency electric fields.

5.1.2 Chemistry (Reaction) Kinetic Engine

The KGMf focuses on chemistry kinetics in global models. One of the least utilized aspects of the KGMf is to use it as a chemistry engine for other simulation codes, e.g., for fluid or other 1D simulation codes (e.g., PIC codes). With a simulation time 0.084s for a single evaluation with the EEDF from a coupled BE solver, the KGMf can be used inside those

codes to recompute species densities from chemical reactions at regular intervals. Using the KGmF as a chemistry engine in other codes, reducing reaction sets in the original simulation code might not be necessary as the KGmF could be used to resolve the reaction kinetics on pre-selected spatial locations. A possible weakness of this setup is in systems with strong gradients of plasma parameters as the non-local effects are not considered, but they affect the EEDF.

The other aspect of the KGmF being utilized as a chemistry engine is to reduce the reaction set on the fly. The KGmF can be utilized as a "stateless" service, where the calling code defines the whole simulation case in the KGmF at every call. That makes it perfect for the KGmF to be utilized as a reaction set reducing engine, reducing the number of species and reactions. With fewer species, the underlying ODE systems are usually smaller, and the whole simulation is faster.

From an implementation viewpoint, coupling with other simulation codes should be an achievable task. With Python being ubiquitous and widely used in the scientific community, coupling with codes in other languages should not be a problem, except in terms of the difficulty of transferring necessary data. One possibility is to run KGmF as a service (Software as a Service, SaaS), which is started at the beginning of the simulation. The simulation code then exchanges the necessary information, i.e., sending parameters to the KGmF and retrieving results from the KGmF, when needed. This setup would reduce the overhead of running the KGmF every time by using more memory in the name of lower overhead.

5.1.3 Interconnected KGmF Codes

The opposite approach to the one described in the previous section is to connect independent global models. Each global model, represented as one KGmF instance, is an independent flow reactor that depends on the output of the previous KGmF instance and provides input for the next KGmF instance. This setup of connected KGmF instances would enable simulation of a multi-reaction (chamber) system, where plasma flows through multiple reactors and is

treated independently in each of them.

Combining the custom transport code and the KGMf would require minor changes in the current KGMf code - one of the main features should be to enable the KGMf to keep "the state of the system" between calls. With this addition, the KGMf would perform a single-time step advance at every call, and the overhead of setting up the ODE system and initializing the system would be removed. The Software as a Service (SaaS) approach is also possible, especially if the KGMf would be an internal service, internal inside the whole transport+KGMf code, and not visible to the operating system.

5.1.4 Energy Dependence in Reaction Set Reduction

Reaction set reduction is an exciting field of investigation. It focuses on describing the chemical kinetics of the given system with a reduced number of species and reactions. Reducing the number of species and reactions can be beneficial in many ways. For example, a smaller number of species directly reduces the size of the ODE system, and a smaller number of reactions make the ODE system less stiff, enabling the use of larger time steps in integration.

Using a smaller ODE system leads directly to the ability to deal with larger reaction sets on the existing hardware and in the same time frame. Reducing the reaction set is vital in plasma-assisted combustion (PAC) systems or systems involving CO₂, where a few hundred species is not unusual, and the number of reactions can be on the order of a few thousand. Reducing the number of reactions affects the stiffness of the ODE system, where integrators can use larger time steps when integrating a less stiff ODE system, leading to shorter integration times.

The main avenue in reaction set reduction is integrating the energy dependence of species and reactions into the reduction workflow. The current reduction method only focuses on defining the importance of species and reactions based on their relative contribution to a specified species' density. The energy aspect of importance is not considered but could be

vital as it could capture the energy transfer between various species states and not only density conversion. Energy-based prioritization could change the order of importance resulting in either a smaller reduced reaction set or just a different reduced reaction set capable of better capturing the behavior of the original reaction set.

5.1.5 Machine Learning for Defining Input Parameters in Sensitivity Analysis

All input parameters for sensitivity analysis presented in this work were defined based on either literature or experiments. In systems with complicated chemistry, many species and reactions present a significant challenge in reducing the number of input parameters for sensitivity analysis. Using machine learning (ML) could help define the importance of individual parameters and define their initial values to best suit the current model.

One possible approach is to use the Gaussian process and Analysis of variance (ANOVA) model to rank input parameters, including reaction rate coefficients and system and regime parameters, such as initial species densities, system volume, power deposition, and initial temperature.

APPENDICES

APPENDIX A

MODELS ADDED TO THE KGFM

High-voltage Collisional Sheath Model in the KGFM

The high-voltage collisional sheath models were added to the KGFM to support approximating the effects of the sheath in nanosecond pulse discharges. Pulsed nanosecond discharges can exhibit very high instantaneous deposited powers, causing the (reduced) electric field to be high as well and making low-voltage sheath models invalid. The sheath model equations were derived from equations in Leiberman and Lichtenberg [98] and adapted to the global model and geometries supported in the KGFM.

In all equations below, V is the volume of the system, and A_{wall} is the area of the wall (electrodes or walls) where the particle and energy losses are computed. When defining the area A for losses, the correct system geometry (rectangular or cylindrical system) is considered.

Equations presented in this section are not tested, and more testing needs to be performed. Preferably, an additional set of tests for testing high-voltage sheath should be added to the KGFM to enable testing high-voltage sheath models to perform correctly at future upgrades.

Charged particles: energy and particle loss

The sheath potential drop (V_s) and Bohm velocity (u_B) both depend on whether the wall is floating or driven/grounded and are computed as:

- floating sheath: $V_s = 0.5T_e \ln\left(\frac{M}{2m_e}\right)$, $u_B = \left(\frac{q T_e}{M}\right)^{1/2}$
- grounded/driven: $V_s \approx 0.78V_0$, $u_B = \left(1 + \frac{\pi\lambda_{Ds}}{2\lambda_i}\right)^{-1/2} \left(\frac{q T_e}{M}\right)^{1/2}$

where m and M are the electron and the ion mass, respectively, T_e is the electron temperature, $\lambda_{Ds} = \left(\frac{\varepsilon_0 T_e}{e n_e}\right)^{1/2}$ is the Debye length in the sheath, $\lambda_i = (\sigma_i n_g)^{-1}$ is the ion mean free path, and V_0 is the amplitude of the voltage source across the gap.

The ratio of particle densities in bulk plasma and the sheath depends on the geometry of the system and can be computed as:

- parallel-plane geometry: $h_l \approx 0.86 \left[3 + \frac{L}{2\lambda_i} \left(\frac{0.86 L u_B}{\pi D_a} \right)^2 \right]^{-1/2}$
- cylindrical geometry: $h_r \approx 0.8 \left[4 + \frac{R}{\lambda_i} \left(\frac{0.8 R u_B}{\chi_{01} J_{01}(\chi_{01}) D_a} \right)^2 \right]^{-1/2}$

where $D_a \approx D_i (1 + T_e/T_i)$ is the ambipolar diffusion with D_i computed by Eq. (A.1), $\chi_{01} = 2.405$, and $J_{01}(\chi_{01})$ is the Bessel function of the first kind of order 0.

$$D_i = \frac{\pi}{8} \lambda_i \bar{v}_i \quad (\text{A.1})$$

where $\bar{v}_i = \left(\frac{8eT_i}{\pi M_i}\right)^{1/2}$ is the ion mean velocity.

With the particle flux to the wall defined as:

$$\Gamma_{l,r} = n_o h_{l,r} u_B \quad (\text{A.2})$$

where u_B is the Bohm velocity, and n_o is the particle density at the center of the system, the particle loss rate at the wall is computed by:

$$R_{l,r} = \Gamma_{l,r} \frac{A_{wall}}{V} \quad (\text{A.3})$$

and the energy losses due to the loss of charged particles to the wall are computed by:

$$\varepsilon_{l,r} = \Gamma_{l,r} \frac{A_{wall} V_s}{V} \quad (\text{A.4})$$

where A_{wall} is the area of the wall, V_s is the voltage drop across the sheath, and h_l and h_r are the density ratios in longitudinal and radial directions, respectively.

Neutral particles: energy and particle loss

The energy and particle loss of neutral particles only considers diffusion losses and neglects all surface processes. Surface processes are neglected, and it is assumed that particles reaching the wall are absorbed according to the probability γ .

The flux of particles A to the wall due to the diffusion in particles B and their absorption in the wall in the system with volume V and wall with area A_{wall} are computed as:

$$\Gamma_{AB} = \frac{1}{A_{wall}} \left[\frac{\lambda_0^2}{D_{AB}} + \frac{(2-\gamma)}{\gamma} \frac{V}{\bar{v}_{AB} 2A_{wall}} \right]^{-1} \quad (\text{A.5})$$

where A_{wall} is the area of the wall, λ_0 is the diffusion length given as $\lambda_0 = L/\pi$ for parallel-plane wall and $\lambda_0 = R/2.405$ for cylindrical wall, D_{AB} is the diffusion of particle A in particle B given by Eq.(A.6), γ is the probability factor of the loss to the wall, $\bar{v}_{AB} = \left(\frac{8eT}{\pi M_R} \right)^{1/2}$ is the average velocity, and $M_R = \frac{M_A M_B}{M_A + M_B}$ is the reduced mass.

The diffusion coefficient of particles A due to collisions with particles B is given by:

$$D_{AB} = \frac{\pi}{8} \lambda_{AB} \bar{v}_{AB} \quad (\text{A.6})$$

where $\lambda_{AB} = (\sigma_{AB} n_B)^{-1}$ is the mean free path between collisions of particle A with particle B. Cross section σ_{AB} is typically in a range $2 - 6 \times 10^{-15} \text{ cm}^2$ and can be estimated from $\sigma_{AB} \approx \pi(r_A + r_B)^2$ via a hard-sphere model, absent of measured cross sections.

The particle loss rate at the wall is computed by:

$$R_{AB} = \Gamma_{AB} \frac{A_{wall}}{V} \quad (\text{A.7})$$

and the energy losses due to the loss of particles A at the wall are computed by:

$$\varepsilon_{AB} = \Gamma_{AB} \frac{A_{wall} V_s}{V} \quad (\text{A.8})$$

where Γ_{AB} is the flux of particles absorbed by the wall, and V_s is the voltage drop across the sheath.

APPENDIX B

COMPUTATIONAL ADDITIONS TO THE KGMF

Testing Suite in the KGMf

Testing is an integral part of software development and regularly testing the program's functionality help keeping the program perform inside the designed specification. Actively developed programs can suffer from introducing new bugs with new code functionality or breaking the existing (and tested) code. Testing can be performed manually or automatically, with manual testing relying on programmers to run tests. Automatic testing can be performed on every commit of the code to the repository, or selected branches in the repository are tested at regular time intervals (overnight tests). Tests are usually designed while designing the code's functionality and focus on assuring the code behaves as designed, e.g., methods should return designed values for known parameters, the correct responses of methods to wrong input parameters, or correctly handling exceptions in the code. The testing suite is composed of many tests, each usually focusing on a particular method or module. Each test includes input parameters and expected values for the method or module. If the return values from the code are the same as expected values, the code passed the test; otherwise, the test is marked as failed.

The initial tests were implemented during the transition from Python2 to Python3 to ensure that the KGMf will still perform as intended. Those tests focused on the main differences between Python2 and Python3, such as different handling of dictionary keys, reading input parameters from NPZ files, creating databases of models and reaction data, and random number generators.

In general, tests are divided into two logical groups that focus on different aspects of the KGMf: unit tests and model tests. While unit tests focus on testing small chunks of

code or individual functions (methods), testing of models focuses on system level testing - treating the KGMf as a black box. In both cases, tests are run, and computed values from the test are compared to expected values stored inside each test directory. Depending on the nature of the test, computed and expected values are compared differently. The computed values need to be either exactly as expected or within the prescribed interval of values for a test to pass. The difference in testing method comes from the nature of the test performed: tested methods in unit tests are (usually) deterministic, and the computed values have to be exact (e.g., loading reaction rates from a file). When running model tests or tests involving iterations, the computed values have to be inside a specific range as some level of uncertainty is present due to the statistical variances of the system.

The testing suite uses `pytest` for performing the tests, and all tests were designed according to the recommendation in the `pytest` documentation. All testing scripts and required files, e.g., input files, required data files, and files with expected results, are stored in the subdirectory `tests` in the main KGMf directory. The structure of directories in the `tests` directory reflects the internal structure of the KGMf to make adding and maintaining the test suit as simple as possible. Running the tests is simplified, and all tests can be run with one command: `pytest -v tests`. The argument `-v` causes verbose output, displaying individual tests performed with current progress (see figure B.1). Other arguments are available by using the switch `-h` (help).

Preparing Expected Values for a Selected Test

All tests are based on the idea that the results from current test runs will be compared to expected values. These expected values are stored either in test scripts, when the number of values is small or in a separate file when the number of testing values is large. When adding a new test to the test suite or when expected test values need to be changed, e.g., due to a bug fix or change in computing algorithm, expected values can be easily saved into a file for later testing.

```

===== test session starts =====
platform linux -- Python 3.8.3, pytest-5.4.3, py-1.10.0, pluggy-0.13.1 -- /home/janez/anaconda3/envs/kgmf/bin/python
cachedir: .pytest_cache
rootdir: /home/janez/work/msu/testing/KGMf_master
collected 31 items

tests/KGM_core/EEDF_solver/test_10_load_Xsec_solver.py::test_lxcat_xsec_old_file_single_bolos PASSED [ 3%]
tests/KGM_core/EEDF_solver/test_10_load_Xsec_solver.py::test_lxcat_xsec_old_file_multiple_bolos PASSED [ 6%]
tests/KGM_core/EEDF_solver/test_20_define_eedf_solver.py::test_no_eedf_solver PASSED [ 9%]
tests/KGM_core/EEDF_solver/test_20_define_eedf_solver.py::test_eedf_solver_no_reactions_involving_be PASSED [ 12%]
tests/KGM_core/EEDF_solver/test_20_define_eedf_solver.py::test_no_eedf_solver_with_reactions_involving_be PASSED [ 16%]
tests/KGM_core/EEDF_solver/test_20_define_eedf_solver.py::test_eedf_solver_defined PASSED [ 19%]
tests/KGM_core/EEDF_solver/test_30_evaluate_eedf.py::test_bolos_direct_init[0.0001] PASSED [ 22%]
tests/KGM_core/EEDF_solver/test_30_evaluate_eedf.py::test_bolos_model_init[0.0001] PASSED [ 25%]
tests/KGM_core/EEDF_solver/test_40_compute_parameters_from_EEDF.py::test_bolos_direct_init[0.0001] PASSED [ 29%]
tests/KGM_core/EEDF_solver/test_40_compute_parameters_from_EEDF.py::test_bolos_model_init[0.0001] PASSED [ 32%]
tests/KGM_exe/model_eval/test_20_gather_reaction_data.py::test_gather_reaction_data PASSED [ 35%]
tests/KGM_exe/model_eval/test_30_prepare_general_ode.py::test_general_ode_system PASSED [ 38%]
tests/KGM_exe/model_eval/test_40_generate_reaction_rates.py::test_gather_reaction_data PASSED [ 41%]
tests/KGM_exe/model_eval/test_50_prepare_specific_ode.py::test_gather_reaction_data[0.0001] PASSED [ 45%]
tests/KGM_exe/model_eval/test_60_integration_time_list.py::test_integration_times_default[0.0001] PASSED [ 48%]
tests/KGM_exe/model_eval/test_60_integration_time_list.py::test_integration_times_one_interval_run[0.0001] PASSED [ 51%]
tests/KGM_exe/model_eval/test_60_integration_time_list.py::test_integration_times_one_interval_run2[0.0001] PASSED [ 54%]
tests/KGM_exe/model_eval/test_60_integration_time_list.py::test_integration_times_interval_dt1[0.0001] PASSED [ 58%]
tests/KGM_exe/model_eval/test_60_integration_time_list.py::test_integration_times_interval_dt2[0.0001] PASSED [ 61%]
tests/KGM_visualize/kgmf_viewer/test_Results_file.py::test_file_load_nonexistent PASSED [ 64%]
tests/KGM_visualize/kgmf_viewer/test_Results_file.py::test_file_load_with_eedf_from_bes PASSED [ 67%]
tests/KGM_visualize/kgmf_viewer/test_Results_file.py::test_file_load_without_bes PASSED [ 70%]
tests/KGMmodels/GP_GRI_Ar_CO2_CH4/test_GP_GRI_Ar_CO2_CH4.py::test_base[0.0001] PASSED [ 74%]
tests/KGMmodels/GP_GRI_Ar_O2_H2/test_GP_GRI_Ar_O2_H2.py::test_base[0.0001] PASSED [ 77%]
tests/KGMmodels/MIXED_GRI_MORGAN_Ar_CO2_CH4/test_mixed_GRI_morgan_Ar_CO2_CH4.py::test_base[0.0001] PASSED [ 80%]
tests/KGMmodels/MIXED_GRI_MORGAN_Ar_O2_H2/test_mixed_GRI_morgan_Ar_O2_H2.py::test_base[0.0001] PASSED [ 83%]
tests/KGMmodels/PLAS_ALL_paper_results/test_plas_ALL_paper_results.py::test_densities[0.0001] PASSED [ 87%]
tests/KGMmodels/PLAS_BIAGI_Ar/test_plasma_biagi_ar.py::test_base[0.0001] PASSED [ 90%]
tests/KGMmodels/PLAS_BIAGI_Ar/test_plasma_biagi_ar.py::test_test1[0.0001] PASSED [ 93%]
tests/KGMmodels/PLAS_BIAGI_Ar/test_plasma_biagi_ar.py::test_reAll[0.0001] PASSED [ 96%]
tests/KGMmodels/PLAS_BIAGI_Ar/test_plasma_biagi_ar.py::test_reaction_info_plas PASSED [100%]

===== 31 passed, 77 warnings in 729.56s (0:12:09) =====

```

Figure B.1 Verbose output of running all tests.

All model tests are designed to use expected values that are stored in the external file and they all use comparison class `CheckComputedValues` (in file `tests/KGMmodel/check_values.py` relative to main KGMf directory) to compare current results with expected values. The class provides two main comparison methods: `checkExact()` to check for exact values, and `checkWithEpsilon()` to check if current values deviate from expected valued less than given relative difference ε .

Tests compare test results acquired with the current version of the KGMf with expected values stored in separate files. Target test results are evolving in time due to the constant development of the KGMf. New tests are added when adding new functionality to the KGMf. In case the expected results of some tests have changed for various reasons (e.g., a new method that replaced the old one, a bug was found and fixed), or a new test was added to the test suit, the expected values can be easily saved into a file for later use. File with expected values can be generated and saved only for an individual test. To save the results of the selected test into a file as reference values, one needs to run the test script as a module and supply the argument `-save_reference` (see example on listing B.1).

Listing B.1 Running a test script and saving values as expected values.

```
1 python -m tests.KGMmodels.PLAS_BIAGI_Ar.test_plasma_biagi_ar --save_reference
```

Before saving expected values for the test into a file, one needs to delete the existing file. This extra step is necessary to prevent accidentally deleting reference values used in the test. Error is displayed if the target file exists when selecting saving new values into the file, together with the name of the file that should be deleted.

Unit Testing

Unit tests are designed to test individual functionalities of the KGMf, such as reading reaction data, creating integration times, executing a coupled Boltzmann equation solver, and creating an ODE system. Individual unit tests are arranged into subdirectories that follow the same hierarchy as the KGMf source code, which simplifies debugging when one or more tests fail.

Unit tests currently in the KGMf testing suite are the following:

- **KGM_core/EEDF_solver**: The directory holds tests for testing necessary steps that are involved in using the EEDF solver as a wrapper around the actual implementation of the Boltzmann equation (BE) solver. The individual tests are the following:
 - loading the cross-section from a supplied file directly by coupled BE solver
 - defining the BE solver and setting its parameters
 - evaluating the EEDF and comparing its shape to the shape of the reference EEDFs. All model parameters, cross-sections, and reduced electric field values are defined programmatically in the KGMf and set in the BE solver.
 - computing electron temperature and mobility by coupled BE solver based on gas mixture and cross-sections defined within the KGMf.

- **KGM_exe/model_eval**: The test is designed to test individual steps needed in creating the model inside the KGMf. The individual steps are tested on a model with pure argon, Maxwellian EEDF, and constant absorbed power. The last test is designed to test the power shape defined by an external function. The individual tests are the following:
 - gathering reaction data from supplied reaction (model) files
 - preparing general ODE system
 - generating reaction rates that are defined as analytical functions
 - preparing specific ODE system
 - generating list of times for integration
 - reading and using power shape (form) defined by an external function
- **KGM_visualize/kgmf_viewer**: The test is designed to test loading of results file with EEDFs from coupled Boltzmann equation (BE) solver and one with analytically defined EEDFs.

Testing of Models

Testing of models focuses on the overall functionality of the KGMf by testing whether results from running the model with the current version of the KGMf still produce correct results. These tests are not concerned with the internals of the KGMf and are treating the KGMf as a black box: specifying the input parameters as input files (simulation and reaction files) and testing the results saved into the NPZ file. The results from individual tests are compared to the expected value - the relative difference should be smaller than $\varepsilon = 10^{-4}$ for a test to pass.

All models in the test directory are supplied with the KGMf. They are automatically generated with the script **quick_generate.py**, which should be run before the first use of

the KGMf. Model input files are stored in the directory `data/KGMmodels/<name of the model>`. The list of models, together with their short description, is given in the list below:

- **GP_GRI_Ar_CO2_CH4**: a gas-phase simulation model with a mixture of argon, CO₂, and CH₄, atmospheric pressure, gas temperature 1200K, 31 species, and 174 reactions
- **GP_GRI_Ar_O2_H2**: a gas-phase simulation model with a mixture of argon, oxygen, and hydrogen, atmospheric pressure, gas temperature 1200K, nine species, and 27 reactions
- **MIXED_GRI_MORGAN_Ar_CO2_CH4**: a plasma simulation model with a mixture of argon, CO₂ and CH₄, atmospheric pressure, gas temperature 500K, Maxwellian EEDF, and a constant absorbed power 10W. The model has 156 species and 330 reactions.
- **MIXED_GRI_MORGAN_Ar_O2_H2**: a plasma simulation model with a mixture of argon, oxygen, and hydrogen, atmospheric pressure, gas temperature 1200K, Maxwellian EEDF, and a constant absorbed power 10W. The model has 60 species and 104 reactions
- **PLAS_ALL_paper_results**: a pure argon gas-phase simulation model to test results from paper by Ashida, Lee and Lieberman[117]
- **PLAS_BIAGI_Ar**: This test includes three very similar plasma simulation models with pure argon, Maxwellian EEDF, atmospheric pressure, and gas temperature of 500, 600, or 1200K. They have either constant or pulsed deposited power, up to 19 species, and up to 64 reactions.

Values tested in model testing in the KGMf are the following:

- sorted list of reactions in the model
- shape (dimensions and sizes) of resulting densities
- values of species densities in the last 5% of simulated time
- times at which densities were computed
- sorted list of variables, which includes all species and temperatures

Sensitivity Analysis Runs with KGMf

Global models are often used to perform the sensitivity analysis of systems due to their capability to capture the main system characteristics and rapidly perform many simulations needed for sensitivity analysis. The KGMf supports performing the sensitivity analysis by defining necessary parameters with a single configuration file and executing individual simulations in parallel to maximize the resources during the sensitivity analysis. The sensitivity analysis run is launched using `run_model_sensitivity.py` script and supplying the necessary parameters with a config file. The sensitivity analysis run mode was motivated by the desire to simplify the investigation of the sensitivity of selected plasma parameters (results) to uncertainty in input parameters. Its functionality is built on top of a batch mode run, using batch runs to run a series of different simulation cases with given parameters. The batch mode can also be invoked independently from the sensitivity analysis run. It enables users to define a set of different cases that are almost the same and differ in a small number of parameters, e.g., different gas pressures or ratio mixtures. Using the batch mode mitigates the possibility of introducing errors compared to manually creating input files and reduces the simulation setup time. After all simulation cases in sensitivity run are performed, the sensitivity analysis run collects the requested results from individual cases and stores them

in a single HDF5 (Hierarchical Data Format 5) file for easy retrieval and post-processing (e.g., plotting or analysis). The HDF5 file format is explained in detail in Section B.

All necessary batch and sensitivity analysis parameters are defined using a single YAML (YAML Ain't Markup Language) input file. The YAML file defines the base simulation and reaction input files (input file templates), together with parameters and values used when creating a list of individual simulation cases. The templates for simulation and reaction input files are the same input files that are used for a single simulation case (usually run with **run_model.py**). The simulation and reaction input files for each simulation case in a batch run (and sensitivity run) are generated automatically. These individual simulation cases are defined according to the settings in the YAML input file and specified simulation and reaction input files (templates). The YAML input file details are presented in section B. The results from individual simulation cases are stored in a separate **numpy** NPZ results file. Input and results files from all individual runs are stored in separate sub-folders, input files in **batch_inputs_XXX** (batch run) and **sa_inputs_XXX** (sensitivity analysis run), and result files in **batch_results_XXX** (batch run) and **sa_results_XXX**. In all folder names, **XXX** is replaced with the case number and tag defined in the input YAML file. The result files from batch and sensitivity analysis runs have the same format as individual KGMf simulation runs.

Defining Parameters and Their Values

Input files for individual simulation cases used in the sensitivity and batch runs are generated from the same simulation and reaction files. They differ only in a small number of system attributes (e.g., volume, pressure, sheath losses, gas mixture) or system parameters (e.g., initial temperature or density, reaction rate coefficients). With that functionality in mind, the parameters in a YAML input file can be arranged into three groups: base input files (templates) and result file, running parameters, and variable parameters and their values.

The base input files define simulation and reaction input files that serve as templates for

all simulations in a batch run. All simulation parameters are first taken from base input files and are changed according to the settings in the YAML file before they are written as input files for a specific case. A unique sequential number (an ID) is assigned to every case within a given batch run, and the simulation case's parameters are saved into the HDF5 file, providing a mapping between a simulation case ID and parameters used in the simulation case. The results file parameter defines the name and the location of the HDF5 file used to store all batch and sensitivity analysis run parameters. These parameters include names and contents of both input files, list of simulation cases, their numbers and parameters used for each case (value and indexes to values), case run status (prepared, input files generated, finished successfully or with error), running time for each case, and the number of done cases.

The second group of parameters in the YAML file holds the running parameters, which define options for running the batch case. They define run times (type and values similarly as for the individual KGMf case), case tag (used in names of directories and files), and the number of cases to run in parallel.

The last group of parameters in the YAML file defines parameters and their values that are changed during the generation of individual simulation cases. The parameters in batch and sensitivity runs can be varied in three different ways:

- keyword `combine`: create all possible combination of values and parameters
- keyword `varysingle`: vary one parameter at a time to utilize elementary method of defining parameter importance(see 3.2)
- keyword `global_sa`: vary parameters according to the Salitelli [76] sampling method to utilize a variance-based method of defining parameter importance (section 3.3)

When keywords `combine` and `varysingle` are present at the same time, the combination of both is used in a way that parameters defined by `varysingle` parameters are repeated for all cases in the `combine` group. The list is used to define values for each parameter

in a group, and all values in the list are used during the creation of cases. The keyword `global_sa` has to be used on its own and is not compatible with the other two parameter groups.

A keyword `combine` can be used to define a group of parameters and their values, where the created list of cases include all possible combinations of all parameters and their values. The example in listing B.2 defines three parameters to vary: `PRESSURE` (4 values), `T_E` (3 values), and `INIT_0` (2 values), each with their own list of values. The total number of cases is $4 \times 3 \times 2 = 24$ different cases.

Listing B.2 Example of `combine` keyword when defining parameters in batch run.

```
1 combine: # create cases with all combination of values
2 PRESSURE: [190, 380, 760, 1000]
3 T_E: [1.5, 2.0, 2.5]
4 INIT_0: [1e+8, 1e+9]
```

The use of `combine` group for defining input parameters can quickly lead to combinatorial explosion, where the number of cases to run increases exponentially with the number of parameters and the number of values for each parameter. This is especially true in a sensitivity analysis, where it is often desirable to know the combined effect of uncertainties in reaction rate coefficients. For example, to check for combined effects of uncertainties in 10 reactions with $\pm 20\%$ uncertainty (defining three values: -20% , 0% , and $+20\%$ uncertainty), it would result in running $3^{10} = 59,049$ individual simulation cases.

A keyword `varysingle` can be used to define a group of parameters that are changed one-at-the-time, presenting an ideal way to create a simulation set for sensitivity analysis using Morris' method [75], which is presented more in detail in section 3.1. The `varysingle` parameters will create a list of cases that will enable studying the individual effect of uncertainty in each parameter on the simulation results. Each parameter is defined with two values: the list of values to use when creating cases and the index of the base value in the list. The index is defined with a keyword `base_index` and is zero-based - the first value in the list has index 0. The `base_index` defines a value in the list that is considered a 'base

value' and is used to defined a 'base case' - the case with case number 0 - and with all parameters having values defined at 'base_index' position. The example in listing B.3 defines cases where reaction rate coefficients for reactions 1 through 4 will be changed according to the given list with 11 values. The base value is value **1.0**, as defined by `base_index: 5` (line 3). This definition of sensitivity analysis would produce $4 \times (11 - 1) + 1 = 41$ individual cases.

Listing B.3 Example of `varysingle` keyword for sensitivity analysis run.

```

1 varysingle:           # vary one reaction coefficient at the time;
2   'R1-R4,R10':       # R1=reaction 1; R1-R4 -> R1,R2,R3,R4; R10
3     base_index: 5    # index of the base value in the list below
4     values: [0.75, 0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.2, 1.25]
5   'R6,R8,R9b':      # R6 (forward and backward),R8,R9 (backward only)
6     base_index: 5    # index of the base value in the list below
7     values: [0.75, 0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.2, 1.25]
8   # combinations are possible
9   'R11-R20,R22,R30-R34':
10    base_index: 5    # index of the base value in the list below
11    values: [0.75, 0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.2, 1.25]
12   'R41-R43b':      # 'b' defines backward reactions for R41 -> R43
13     base_index: 5    # index of the base value in the list below
14     values: [0.75, 0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.2, 1.25]

```

A keyword `global_sa` can be used to define a group of parameters whose values are changed according to the Saltelli [76, 104] sequencing to capture the combined effects of uncertainties in input parameters to output results. The number of required cases using Saltelli sampling is significantly smaller compared to the naïve approach offered by the `combine` parameter definition. The sampling aims to achieve the best possible coverage of input parameter space with fewer individual cases. The values for each parameter are automatically generated according to the given distribution (e.g., normal, lognormal, gamma) and given distribution parameters (e.g., mean, mode, variance, scale). The total number of generated cases depends on the number of input parameters (p) and the number of samples for each parameter (M) and is $M(2p + 2) + 1$. For example, the listing B.4 presents a setting of sensitivity analysis parameters with $p = 10$ input parameters (reactions 1 through 10) and $M = 10$ samples for each parameter (the second parameter for the lognormal distribution). The total number of cases is $10(2 \cdot 10 + 2) + 1 = 221$, which is much smaller compared to

10^{10} cases when using a naïve approach with testing all possible values of each parameter.

Listing B.4 Example of `global_sa` keyword for sensitivity analysis run.

```
1 global_sa:      # perform global sensitivity analysis
2   # log-normal distribution: 10 samples, mode=1.0 (the value with the
3   # most probability), sigma=0.2
4   'R1-R10': lognormal 10 1.0 0.2
```

Regardless of the type of parameter group used, there are two main differences when defining reaction rate coefficients and other parameters:

- reaction definitions always start with an uppercase letter **R**, followed by the reaction number, e.g., R999, where 999 is the reaction number (reaction numbers are 1-based - the first reaction has number 1), while other parameters have to be defined with the exact name as it is written in the input files.
- the values in the YAML file for reactions are specified as relative numbers that are multiplied with reaction rate coefficients already defined in the reaction input file, while absolute values have to be specified for all other parameters.

The main reason for using relative values for reaction rate coefficients is the flexibility in defining the uncertainties of the reaction rate coefficients without knowing their original (unmodified) values.

Collecting Results for Sensitivity Analysis

Sensitivity analysis runs consist of many individual simulation cases with input parameters generated in a prescribed way. The input files for the individual case are automatically generated, and results for each case are saved into individual results files. The results from individual simulation cases are also collected into a single HDF5 file in the form of multidimensional **numpy** arrays for easy post-processing through a wide variety of available tools. The shape of stored results, i.e., the number of dimensions and their sizes, depends on the parameter definitions in the YAML input file (lines 2-10 in listing B.5). In the case

of the `combine` group, each parameter adds one dimension to the array. In the case of the `varysingle` group, the array will have two dimensions: the number of rows is defined by the number of parameters, and the number of columns is defined by the length of the longest list of parameters. In the case of the `global_sa` parameter group, two dimensions are created in the array: the number of rows is defined by the number of created cases and the number of columns by the number of reactions. The last dimension in the array is always defined by the number of values collected in time (e.g., 20 values are defined on line 18 in listing B.5).

Listing B.5 Example of YAML file for collecting results.

```

1 parameters:
2   case1: # name of the case (used below)
3     combine: # defined first 2 dimensions of 3 x 2 elements
4       T_E: [1.5, 2.0, 2.5]
5       INIT_0: [1e+8, 1e+9]
6     varysingle: # define next two dimensions..
7       'R1-R4': # .. with size 4 (4 reactions)
8         base_index: 3
9         # .. and size 7 (7 values in the list)
10        values: [0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15]
11 collect:
12   - case_name: case1 # from which case to collect values
13     hdf5group: last_20 # the HDF5 group needs to be unique
14     variables: # for which variables to collect the values
15       - Electron
16       - Ar_pion_0
17     time_slice: # the last dimension is time..
18       last: 20 # .. with 20 elements (20 times steps)

```

Listing B.5 shows a minimal configuration for sensitivity analysis with mixed parameter definitions (`combine` and `varysingle` parameter groups) and collecting results of two variables (electron and Ar ion densities) computed in the last 20 time steps. The dataset of every collected variable, i.e., `Electron` (electrons) and `Ar_pion_0` (argon ions), would have 5 dimensions with the length of dimensions being 3, 2, 4, 7, and 20 elements and resulting in $3 \times 2 \times 4 \times 7 \times 20 = 3360$ elements for each variable (see Table B.1).

Each YAML input file can include definitions for collecting multiple results. Each definition needs to have a unique name (line 13 in listing B.5) that is also used as an HDF5 group name. Using unique names enables different collections from the same results (e.g., defining different time intervals) or grouping the result values, e.g., neutral particles, charges particles, and temperatures. The process of collecting the results can be done many times

Table B.1 Dimensions and their sizes when storing collected results in sensitivity analysis example presented in listing B.5.

Dimension	Length	Explanation
1	3	number of values for parameter T_E
2	2	number of values for parameter INIT_0
3	4	number of reactions: R1-R4
4	7	number of values for each reaction
5	20	number of values at different times
	3360	total number of values stored

after the simulations are done by invoking the `run_model_sensitivity.py` script with the additional argument `c` (collect results only). One can also modify the definition of results to be collected in the YAML file or add new definitions of results to be collected and rerun the result collecting.

Input File for Batch and Sensitivity Analysis Runs

Batch run and sensitivity runs in KGMf are defined with parameters from a supplied YAML file. YAML input file defines all necessary parameters for conducting a batch run and variables for collecting during the sensitivity analysis. As described before, the sensitivity run is based on a batch run with the YAML file structure reflecting that, and the same YAML file can be used for batch and sensitivity runs.

Listing B.6 shows an example for YAML file. The YAML file can be divided into two logical parts: batch run parameters and parameters for sensitivity analysis collection. The main parameters for the batch run are names of input files (lines 2-4), run times (line 7), case name (line 13), and variables to vary and their values (lines 14-20). The main parameters for sensitivity analysis collection are grouped under `collect` tag (line 21) and define one or more HDF5 groups (line 23) into which collected results are stored. Each group has to include: (a) a name of the case from which to get results (line 22), (b) a list of variables for which to collect values (lines 25-27), and (c) times at which to collect values (line 29).

The list of variables can hold only variables defined in the system (in the simulation or reaction input files). A file can have multiple collect sections for one simulation case (`case1` in the example), with unique `hdf5group` name (lines 23 and 31) and different time range or variables to collect.

Listing B.6 Example of YAML file used for batch and sensitivity runs.

```

1 input_files:           # define main input files
2 simulation_file: SI_base.txt
3 reaction_file: RF_base.txt
4 resultsFile: results_sa_Ar_base_yaml.hdf5
5 tag: base             # optional: general tag (used in directories)
6 run_times:           # run times - type 3, dt=1e-12 from 0 till 1e-7s
7   3: 1e-7 1e-12
8 # optional; number of CPUs to use, if not given, 1 is used
9 numCPU: 10
10 # mandatory; parameters for running
11 # this are paramters to define cases
12 parameters:
13   case1:              # name of the case (used below)
14     combine:          # create cases with all combinations of values
15       T E [1.5, 2.0, 2.5]
16       INIT_0: [1e+8, 1e+9]
17     varysingle:      # vary one reaction coefficient at the time;
18       'R1-R4':       # R1=reaction 1; R1-R4 -> R1,R2,R3,R4
19       base_index: 5  # index of the base value in the list below
20       values: [0.75, 0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.2, 1.25]
21 collect:
22   - case_name: case1  # from which case to collect values
23     hdf5group: last_20 # the HDF5 group needs to be unique
24     variables:        # for which variables to collect the values
25       - Electron
26       - Ar_pion_0
27       - Ar2_pion_0
28     time_slice:       # for which times
29       last: 20        # collect values for the last 20 time steps
30
31   - case_name: case1  # from which case to collect values
32     hdf5group: exc_range # the HDF5 group needs to be unique
33     variables:        # for which variables to collect the values
34       - Ar_exc_0
35       - H_exc_1
36       - H_vib_0
37     time_slice:       # for which times
38       range: 1e-8 1e-7 # collect values between given times

```

For the purpose of global sensitivity analysis with variance-base parameter values selection (section 3.3), there is also the third possibility to define parameter variations: `global_sa` (see listing B.7, line 14). The `global_sa` parameter definition is not compatible with previous parameters groups. Parameters are defined in a similar way as before, the values for each parameter are define as follows (line 16 on listing B.7): `'R1-R10': lognormal 200 1.0 0.05`, where **200** is the number of samples to drawn from the **lognormal** distribution

with mode **1.0** (the most probable value from the distribution), and standard deviation **0.2**. The distribution mode and the standard deviation are parameters for the distribution of multiplication factors for reaction rate coefficients (or cross-sections) used to compute (define) rates for each case. The base case will always have a multiplication factor of 1.0. Currently, only normal and log-normal distributions are supported. The samples are generated (drawn) using **SALib** library.

Listing B.7 Example of YAML file used for sensitivity runs (`global_sa`).

```

1 # define main input files
2 input_files:
3   simulation_file: SI.txt
4   reaction_file: RF.txt
5   resultsFile: results_sa_200samples.hdf5
6 tag: 200samples      # optional: general tag (used in directories)
7 run_times:          # run times; run_type: times arguments
8   3: 1e-15 1e-17 1e-4 -1e-3 1e-3 1e-6
9 # optional; number of CPUs to use, if not given, 1 is used
10 numCPU: 12
11 # mandatory; parameters for running
12 parameters:
13   base:
14     global_sa:      # perform global sensitivity analysis
15                   # normal distribution, 200 samples, mean=1.0, std.dev=0.2
16     'R1-R10': lognormal 200 1.0 0.2
17 collect:
18   - case_name: base
19     hdf5group: 200samples
20     variables:      # observed variable(s)
21       - Ar_g
22       - Ar_pion_0
23       - Ar2_pion_0
24       - Ar_exc_m
25       - Electron
26     time_slice:    # times to store
27     last: 200

```

The KGMf Results File Format

The KGMf offers three different running modes that can save results in two different file formats. In the case of running a single simulation case via a `run_model.py` script, the results are stored in NPZ (compressed **numpy**) file. In the case of running batch or sensitivity analysis run via `run_model_batch.py` or `run_model_sensitivity.py` scripts, the results are stored into HDF5 (Hierarchical Data Format 5) file format. The results stored in **numpy**

NPZ files can be viewed using Python script `view_results.py` that is part of the KGMf. There is no script (viewer) added to the KGMf that would offer to view the results from batch and sensitivity analysis runs. A short Python script included in this document will read the HDF5 file and print out the skeleton (hierarchy) of the HDF5 file, enabling users to extract required data from HDF5 files.

numpy NPZ File Format

The **numpy** NPZ file format is used to store the results from a KGMf's single case simulation run. The file contains some results even if the simulation run fails. There are many sections (variables) in the results file. Some are always present, and some are optional, depending on either KGMf's running parameters or the simulation type used to save the results. The following list consists of sections that are always present in the NPZ file:

- *t_vals* [**numpy** array]: A list of times for which the results (densities and temperatures) are stored in the results file. The order of time values correspond to rows in the *densities* array.
- *densities* [**numpy** array]: A **numpy** array storing results from the simulation: rows corresponds to one time value (order given by *t_vals*) and columns are variables (order given by *var_list*).
- *last_integration_time* [float]: The last simulation time for which results are available in the file. In most cases, this is the same time as the computational end time (last value in *t_vals*) but could be different if the simulation failed to finish successfully.
- *var_list* [dictionary]: A list of variables whose results are available in *densities*; the order of variables defines the order of results in columns in the array *densities*.
- *calls_to_integrator* [integer]: The number of calls to the integrator; the list depends on the integrator used in the simulation (e.g., ode or odeint). The value is informative

and is not used in any calculation. It helps evaluate whether the EEDF evaluation frequency was aligned with one selected in the input file.

- *simfile* [string]: file name of the simulation file
- *reactfile* [string]: file name of the reaction file
- *runtime* [float]: total (wall) simulation runtime
- *helper_vals_history* [dictionary]: History of values used to determine if the EEDF needs to be evaluated or not. The field is only populated when the EEDF is computed using coupled BE solver. The data stored is the following:
 - *column_names* [**numpy** array]: The order of names of variables stored in corresponding column in *values*, such as *t* (time), *EN* (reduced electric field), *E* (electron density), and *Pabs* (absorbed power density).
 - *values* [**numpy** array]: The array of values of stored variables. Each row holds values for one time step and columns are given by *column_names*. A time values are stored in the first column.
- *reaction_list* [**numpy** array]: The list of reactions as they are defined in the KGMf; unless something is wrong, this should be the same list as in the reaction input file.
- *K_values* [**numpy** array]: The time evolution (history) of computed reaction rate coefficients; the values stored here are computed separately from the main simulation. It is possible to run the simulation and have the history of different reaction rates computed and stored in the array.
- *K_values_symbols* [**numpy** array]: Symbols for individual reaction rate coefficients as they were used in symbolic representation of the ODE system.

- *power* [**numpy** array]: The history of absorbed power density used in the simulation. The absorbed power is often the input parameter for the simulation, and this array holds absorbed power density values at times defined in *t_vals*.
- *steadyState* [dictionary]: Holds the information about the system reaching a steady state. The steady state condition is computed based on values of electron temperature (T_e) and electron density (n_e). The content of the dictionary is as follows:
 - *tol* [float]: the tolerance used whether determine of the system reached the steady state
 - *clusterSize* [integer]: the number of values over which the steady state was checked
 - *T_e* [dictionary]: was the steady state reached regarding electron temperature ('achieved': True/False) and the value of electron temperature ('value')
 - *el* [dictionary]: was the steady state reached regarding electron density ('achieved': True/False) and the value of electron density ('value')
- *version* [string]: version of the results file

The following sections are saved into **numpy** NPZ file only if the EEDF information should be saved into results:

- *t_vals_eedf_keys* [**numpy** array]: A list that connects the EEDF evaluations and simulation times in the ODE integrator since EEDF evaluations could be done at various times during the simulation. The length of this list is the same as the length of *eedf_data*. The value in the list is the index of the time for which the EEDF was used. The same index of time can be found multiple times in this list - this is the case when the same EEDF was used on multiple occasions when no EEDF evaluation was required.
- *eedf_data* [dictionary]: This Python dictionary stores detailed data of evaluated EEDFs during the simulation. The number of calls to the integrator at the time of saving the

data into the dictionary is used as a key in the dictionary. The data stored for each evaluated EEDF depends on the underlying Boltzmann equation (BE) solver and is the following:

- *t* [float]: time at which the EEDF was evaluated (computed)
- *f* [**numpy** array]: values of function on energy grid given by *energy_grid_centers*
- *energy_grid_centers* [**numpy** array]: cell centers of energy grid
- *energy_grid_centers_odd* [**numpy** array]: cell centers of odd energy grid [67]; This is currently only used by MultiBolt BE solver.
- *num_iter* [integer]: the number of iterations needed to compute the EEDF by the underlying BE solver
- *error* [float]: the error in a computed EEDF as it is returned by the underlying BE solver
- *f_init* [**numpy** array]: values of the EEDF that were used by BE solver as initial values for computing the EEDF
- *solution_converged* [bool/integer]: did the solution of BE solver converged
- *params* [dictionary]: parameters sent to the BE solver to compute stored EEDF
- *Te_from_BES* [float]: electron temperature computed from stored EEDF
- *mue_N* [float]: value of electron mobility computed from stored EEDF
- *eedf_eval_frequency_params* [dictionary]: History of evaluation frequency parameters, i.e., parameters used to decide if the EEDF needs to be re-evaluated or if the current EEDF can be used. It is helpful during the debugging process and has very little use to end-users.

HDF5 File Format

Results from batch and sensitivity runs are saved into a single HDF5 file that enables easy retrieval of results. Each batch and sensitivity run is composed of many independent simulation runs, and the results of each simulation run are stored in a separate **numpy** NPZ file, described in detail in section B. In general, the data stored into HDF5 file can be divided into two parts, one saved by batch run and the other by sensitivity run, and the data saved is as follows:

- **batch run:**

In a batch run, the HDF5 file holds the list of all cases run in this batch run, together with the filename of the detailed result file (**numpy** NPZ file) and the values of all parameters used in every case in a batch run. The list enables the connection between chosen values of input parameters and batch run case numbers.

- **sensitivity run:**

The sensitivity run extends the functionality of a batch run. Along with data saved in a batch run, the sensitivity run stores gathered data for each defined case: the list of target parameters (e.g., temperature or species' density) and the sensitivity of each parameter to the values of input parameters. Values of target parameters are organized according to the input parameter values, enabling easy retrieval of parameter's sensitivity for postprocessing, e.g., plotting of sensitivity analysis.

The listing B.8 shows short Python code that reads and prints out the main hierarchy of the HDF5 file. The script `hdf5_info.py`, located in the KGMf's root directory, will take the HDF5 file name as an argument and print the main skeleton (hierarchy) of the supplied HDF5 file. Sections **params** and `<case_name>` are generated during the batch run, the section **collect** is generated during the sensitivity analysis run.

Listing B.8 Simple Python code to print the hierarchy of the HDF5 file.

```

1 resultFile = '<result_file>'
2 fp = h5py.File(resultFile, 'r')
3
4 if 'params' in fp.keys():
5     grp = fp['params']
6     cases_list = grp.attrs.keys()
7
8     if case_name in cases_list:
9         for key, val in grp.attrs.items():
10            val = json.loads(val)
11            if type(val) == dict:
12                print(print_dict(key, val, 0))
13            else:
14                print(key, ":", val)
15
16 if case_name in fp.keys():
17     grp = fp[case_name]
18     print("Attributes:")
19
20     for key, val in grp.attrs.items():
21         if key != 'cargo':
22             print(" - {0}: {1}".format(key, val))
23         else:
24             vals = json.loads(val)
25             print(" - {0}: {1}".format(key, list(vals.keys())))
26
27     print("Datasets:")
28
29     for key, val in grp.items():
30         print(" - {0}: {1}".format(key, val))
31
32     print("Case numbers with base reaction coefficients:")
33     for i, row in enumerate(grp['cases']):
34         if not np.any(row[1:] - 1):
35             print(" {0}: {1}".format(i, row))
36
37 fp.close()

```

The sections in the HDF5 results file are created either during the batch run (labeled with [batch]) or during the sensitivity analysis run (labeled with [sa]) and are the following:

- *params* ([batch]): The HDF5 group holds parameters for all batch runs saved in the file. The names of batch runs are stored in the attributes of the group. For each case in the group, there are two possible subgroups (keys): *varysingle* and *combine*. Each group holds the list of parameters and the list of their values used to create individual simulation cases.
- *<CaseName1>* ([batch]): The *<CaseName1>* is a HDF5 group with the name of the actual case defined in the YAML input file. The name has to be unique for

a given HDF5 file. Data for each case is stored either in **attributes** (general case information), **datasets** (array oriented input parameters with their values), and **HDF5 group** (collected results).

Attributes

Input parameters that define batch run are stored as attributes to each batch run case and consist out of the following:

- *cases_done*: the number of simulation cases done (simulation has finished)
- *cases_number*: the number of simulation cases attempted in a batch run
- *status*: the description of possible status a case can have; the content of the field is usually: '0-init value, 10-input files created, 20-running, 90-done successfully, 91-error'
- *vars_list*: the list of variables that were used to create different cases in a batch run; the value is created (expanded) from the input parameter definitions in YAML file
- *reaction_file_template_name*: the name of reaction file template used to create the reaction file for each case
- *reaction_file_template_content*: the content of the reaction file template
- *simulation_file_template_name*: the name of the simulation file template used to create the simulation file for each case
- *simulation_file_template_content*: the content of the simulation file template

Datasets

Datasets are used to store parameter values for individual simulation cases, the status of individual runs, and the final results file. Datasets are organized as arrays with dimensions related to the number of individual simulation cases (labeled as M) and

the number of parameters used in a given batch run (labeled as L). All datasets are populated during the batch run, except dataset **results**, which is populated during the sensitivity analysis run. Datasets defined for each case are the following:

- *cases*: a **numpy** array of a size $M \times L$ used to store values of input parameters in each simulation case
- *cases_params_indexes*: a **numpy** array of a size $M \times L$ used to store indexes to values (from list defined in YAML file) of input parameters in each simulation case
- *result_files*: a list of size M that holds the filenames of results files for each (individual) simulation case (**numpy** NPZ file)
- *run_time*: a list of size M that holds the total running time of each simulation case
- *status*: a list of size M that holds the current status of individual simulation case; this status is changing during the batch run and would show the last good status of the simulation case

HDF5 group ([sa])

All collected results are stored in an HDF5 group named **results** with sub-groups names as defined in the YAML file (see Section B for detailed explanation of how collected results are specified). All results have the same structure; they only differ in the shape (size) of the result datasets as the size of arrays depends on the number of variables, selected parameters and their values, and the number of times values stored in the collected results. The structure of the collected results is as follows:

- $\langle GroupName1 \rangle$: The name of the group as defined in the YAML file (line 23 in listing B.6) and has to be unique for each YAML file. Datasets stored in groups are named after variables whose values are stored in the datasets and have the same

size (shape). The shape and size of results dataset depends how parameters are combined (`varysingle`, `combine`, or `global_sa`) and the number of time values stored in the dataset depends on parameter definition used. Parameter definition type `global_sa` is not compatible with other two definitions. The result dataset shape and size is computed in the following way:

- * using either (or both) `varysingle` and `combine`: the number of parameters in `combine` section defines the first number of dimensions of the dataset; then two dimensions are added if `varysingle` parameters are defined, and the last dimension is added to accommodate for time dependence of stored values. The number of values in each dimension is defined by the number of different parameter values in a given dimension. When reading results into Python, datasets behave as **numpy** arrays, and array slicing can be used to extract required values for post-processing.
- * using `global_sa`: results have only two dimensions - the number of rows corresponds to the number of cases (defined by varying parameters and according to Saltelli's sampling), and the number of columns is equal to the number of values collected in time.

For example in listing B.9, the dataset of every collected variable would have 5 dimensions with length of dimensions being 3, 2, 4, 7, and 20 elements and resulting in $3 \times 2 \times 4 \times 7 \times 20 = 3360$ elements for each variable. Individual dimensions and their sizes are defined as follows:

- * due to the `combine` definition, two dimensions for parameters `T_E` and `INIT_0` of sizes 3 and 2, respectively, are added
- * then due to the `varysingle` definitions, two dimensions for reactions `R1-R4` and `values` list of sizes 4 and 7, respectively, are added
- * the last dimension of length 20 is added to store 20 values collected in time

Listing B.9 Snipped of YAML input file (not complete).

```
1 . . .
2 combine:          # defined first 2 dimensions of 3 x 2 elements
3   T_E: [1.5, 2.0, 2.5]
4   INIT_0: [1e+8, 1e+9]
5 varysingle:      # define next two dimensions..
6   'R1-R4':       # .. with size 4 (4 reactions)
7     base_index: 3 # .. and size 7 (7 values in the list)
8     values: [0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15]
9 . . .
10 time_slice:     # the last dimension is time..
11   last: 20      # .. with 20 elements (20 times steps)
```

- * *<GroupName2>*: the results group can hold multiple datasets with unique names and their shapes and sizes according to selected parameters.
- *<CaseName2>* ([batch]): The HDF5 file can store multiple different cases defined by the same YAML file. The structure of data for every case is the same as presented above (under *<CaseName1>*).

APPENDIX C

REACTION SET FOR GLOBAL SENSITIVITY ANALYSIS

Reaction Set for Plasma Discharge in H₂-O₂-Ar Gas Mixture

The following table shows the reaction set for plasma discharge in H₂-O₂-Ar gas mixture, used in the global sensitivity analysis (section 3.4.3) and reaction set reduction (section 4.3).

The reaction set consists of 62 species and 146 reactions.

Table C.1 Reactions used in the plasma discharge in argon with their reaction rate coefficients and threshold energies. Additional subscripts v , r , and e indicate vibrational, rotational and excited states, respectively. *Notes:* T_e is the electron temperature, in eV, T_{gas} is the gas temperature, in K.

Reaction	Rate coefficient [m ³ /s or m ⁶ /s]	Refs.
R1 Ar + e ⇒ Ar + e	cross section	[90]
R2 Ar + e ⇌ Ar* + e	cross section	[90]
R3 Ar + e ⇌ Ar ^{3*} + e	cross section	[90]
R4 Ar + e ⇒ Ar ⁺ + 2e	cross section	[90]
R5 O + e ⇒ O + e	cross section	[90]
R6 O + e ⇌ O(¹ D) + e	cross section	[90]
R7 O + e ⇌ O(¹ S) + e	cross section	[90]
R8 O + e ⇒ O ⁺ + 2e	cross section	[90]
R13 O ⁻ + e ⇒ O ⁻ + e	cross section	[90]
R14 O ⁻ + e ⇒ O + 2e	cross section	[90]
R15 O ₂ + e ⇒ O ₂ + e	cross section	[90]
R16 O ₂ + e ⇌ O _{2,r=1} + e	cross section	[90]
R17 O ₂ + e ⇌ O _{2,v=1} + e	cross section	[90]
R18 O ₂ + e ⇌ O _{2,v=1} + e	cross section	[90]
R19 O ₂ + e ⇌ O _{2,v=2} + e	cross section	[90]
R20 O ₂ + e ⇌ O _{2,v=2} + e	cross section	[90]
R21 O ₂ + e ⇌ O _{2,v=3} + e	cross section	[90]
R22 O ₂ + e ⇌ O _{2,v=4} + e	cross section	[90]
R23 O ₂ + e ⇌ O ₂ (a ¹ Δ) + e	cross section	[90]
R24 O ₂ + e ⇌ O ₂ (b ¹ Σ) + e	cross section	[90]
R25 O ₂ + e ⇌ O _{2,e=3} + e	cross section	[90]
R26 O ₂ + e ⇌ O _{2,e=4} + e	cross section	[90]
R27 O ₂ + e ⇌ O _{2,e=5} + e	cross section	[90]
R28 O ₂ + e ⇌ O _{2,e=6} + e	cross section	[90]
R29 O ₂ + e ⇌ O _{2,e=7} + e	cross section	[90]

continuing on the next page...

... continued from the previous page

Reaction	Rate coefficient [m ³ /s or m ⁶ /s]	Refs.
R30	$O_2 + e \Rightarrow O_2^+ + 2e$	cross section [90]
R31	$O_2 + e \Rightarrow O_2^-$	cross section [90]
R32	$O_2(a^1\Delta) + e \Rightarrow O_2(a^1\Delta) + e$	cross section [90]
R33	$O_2(a^1\Delta) + e \Rightarrow O + O(^1D) + e$	cross section [90]
R34	$O_2(a^1\Delta) + e \Rightarrow O_2^+ + 2e$	cross section [90]
R35	$O_2(a^1\Delta) + e \Rightarrow O + O^-$	cross section [90]
R36	$O_2(b^1\Sigma) + e \Rightarrow O_2(b^1\Sigma) + e$	cross section [90]
R37	$O_2(b^1\Sigma) + e \Rightarrow O + O + e$	cross section [90]
R38	$O_2(b^1\Sigma) + e \Rightarrow O_2^+ + 2e$	cross section [90]
R39	$O_2(b^1\Sigma) + e \Rightarrow O + O^-$	cross section [90]
R40	$O_3 + e \Rightarrow O_3 + e$	cross section [90]
R41	$O_3 + e \Rightarrow O_3^+ + 2e$	cross section [90]
R42	$O_3 + e \Rightarrow O_2 + O^-$	cross section [90]
R43	$O_3 + e \Rightarrow O + O_2^-$	cross section [90]
R44	$H + e \Rightarrow H + e$	cross section [90]
R45	$H + e \Leftrightarrow H_{e=2} + e$	cross section [90]
R46	$H + e \Leftrightarrow H_{e=3} + e$	cross section [90]
R47	$H + e \Rightarrow H^+ + 2e$	cross section [90]
R50	$H_2 + e \Rightarrow H_2 + e$	cross section [90]
R51	$H_2 + e \Leftrightarrow H_{2,r=1} + e$	cross section [90]
R52	$H_2 + e \Leftrightarrow H_{2,r=2} + e$	cross section [90]
R53	$H_2 + e \Leftrightarrow H_{2,v=1} + e$	cross section [90]
R54	$H_2 + e \Leftrightarrow H_{2,v=2} + e$	cross section [90]
R55	$H_2 + e \Leftrightarrow H_{2,v=3} + e$	cross section [90]
R56	$H_2 + e \Leftrightarrow H_{2,e=1} + e$	cross section [90]
R57	$H_2 + e \Leftrightarrow H_{2,e=2} + e$	cross section [90]
R58	$H_2 + e \Leftrightarrow H_{2,e=3} + e$	cross section [90]
R59	$H_2 + e \Leftrightarrow H_{2,e=4} + e$	cross section [90]
R60	$H_2 + e \Leftrightarrow H_{2,e=5} + e$	cross section [90]
R61	$H_2 + e \Leftrightarrow H_{2,e=6} + e$	cross section [90]
R62	$H_2 + e \Leftrightarrow H_{2,e=7} + e$	cross section [90]
R63	$H_2 + e \Leftrightarrow H_{2,e=8} + e$	cross section [90]
R64	$H_2 + e \Leftrightarrow H_{2,e=9} + e$	cross section [90]
R65	$H_2 + e \Leftrightarrow H_{2,e=10} + e$	cross section [90]
R66	$H_2 + e \Rightarrow H_2^+ + 2e$	cross section [90]
R67	$H_{2,e=10} + e \Rightarrow H_{2,e=10} + e$	cross section [90]
R68	$H_{2,e=10} + e \Rightarrow H + H^+ + 2e$	cross section [90]
R69	$H_{2,e=8} + e \Rightarrow H_{2,e=8} + e$	cross section [90]
R70	$H_{2,e=8} + e \Rightarrow H + H^+ + 2e$	cross section [90]
R71	$H_2O + e \Rightarrow H_2O + e$	cross section [90]
R72	$H_2O + e \Leftrightarrow H_2O_{v=1} + e$	cross section [90]
R73	$H_2O + e \Leftrightarrow H_2O_{v=2} + e$	cross section [90]
R74	$H_2O + e \Leftrightarrow H_2O_{v=2} + e$	cross section [90]
R75	$H_2O + e \Rightarrow O(^1D) + H_2 + e$	cross section [90]

continuing on the next page...

... continued from the previous page

Reaction	Rate coefficient [m ³ /s or m ⁶ /s]	Refs.
R76 H ₂ O + e ⇒ H ₂ O ⁺ + 2e	cross section	[90]
R77 H ₂ O + e ⇒ H ₂ + O ⁻	cross section	[90]

Ar, Ar⁺, Ar*, Ar₂⁺ reactions

R106 Ar ₂ ⁺ + e ⇒ Ar + Ar	8 × 10 ⁻¹³	[118]
R107 Ar ⁺ + e + e ⇒ Ar + e	10 ⁻³¹ · (11605 T _e /300) ^{-4.5}	[33]
R108 Ar ⁺ + 2Ar ⇒ Ar ₂ ⁺ + Ar	2.25 × 10 ⁴³ · (T _g /300.0) ^{-0.4}	[119]
R109 Ar* + 2Ar ⇒ Ar + 2Ar	1.4 · 10 ⁻⁴⁴	[119]
R110 Ar ⁺ + H ₂ ⇒ Ar + H ₂ ⁺	2.7 × 10 ⁻¹⁶	[120]
R111 Ar + H ₂ ⁺ ⇒ Ar ⁺ + H ₂	1.461 × 10 ⁻¹⁶	[120]
R112 Ar* + H ₂ ⇒ Ar + H + H	6.6 × 10 ⁻¹⁷	[121]
R113 Ar* + O ₂ ⇒ Ar + O + O	2.1 × 10 ⁻¹⁶	[33]
R146 Ar* + Ar ⇒ Ar + Ar	2.3 × 10 ⁻²¹	[33]

H reactions

R83 H + O ₂ + M ⇌ HO ₂ + M	7.721 × 10 ⁻⁴² T _g ^{-0.86}	[91]
R84 H + 2O ₂ ⇌ O ₂ + HO ₂	5.735 × 10 ⁻⁴¹ T _g ^{-1.24}	[91]
R85 H + O ₂ + H ₂ O ⇌ H ₂ O + HO ₂	3.105 × 10 ⁻⁴¹ T _g ^{-0.76}	[91]
R86 H + O ₂ + AR ⇌ AR + HO ₂	1.930 × 10 ⁻⁴² T _g ^{-0.8}	[91]
R87 H + O ₂ ⇌ O + OH	4.400 × 10 ⁻¹⁴ T _g ^{-0.6707} exp(-8575/T _g)	[91]
R88 2H + M ⇌ H ₂ + M	2.757 × 10 ⁻⁴² T _g ⁻¹	[91]
R89 2H + H ₂ ⇌ 2H ₂	2.482 × 10 ⁻⁴³ T _g ^{-0.6}	[91]
R90 2H + H ₂ O ⇌ H ₂ + H ₂ O	1.654 × 10 ⁻⁴⁰ T _g ^{-1.25}	[91]
R91 H + OH + M ⇌ H ₂ O + M	6.066 × 10 ⁻³⁸ T _g ^{-2.0}	[91]
R92 H + HO ₂ ⇌ O + H ₂ O	6.592 × 10 ⁻¹⁸ exp(-338/T _g)	[91]
R93 H + HO ₂ ⇌ O ₂ + H ₂	7.439 × 10 ⁻¹⁷ exp(-537/T _g)	[91]
R94 H + HO ₂ ⇌ 2OH	1.395 × 10 ⁻¹⁶ exp(-320/T _g)	[91]
R95 H + H ₂ O ₂ ⇌ H ₂ + HO ₂	2.009 × 10 ⁻²³ T _g ² exp(-2617/T _g)	[91]
R96 H + H ₂ O ₂ ⇌ OH + H ₂ O	1.661 × 10 ⁻¹⁷ exp(-1812/T _g)	[91]
R48 H _{e=2} ⇌ H	6.26 × 10 ⁸	[90]
R49 H _{e=1} ⇌ H	2.50 × 10 ⁻⁶	[90]

HO₂ reactions

R103 2HO ₂ ⇌ O ₂ + H ₂ O ₂	2.159 × 10 ⁻¹⁹ exp(820/T _g)	[91]
R104 2HO ₂ ⇌ O ₂ + H ₂ O ₂	6.974 × 10 ⁻¹⁶ exp(-6039/T _g)	[91]

continuing on the next page...

... continued from the previous page

Reaction	Rate coefficient [m ³ /s or m ⁶ /s]	Refs.
O reactions		
R78 2O + M ⇌ O ₂ + M	$3.309 \times 10^{-43} T_g^{-1}$	[91]
R79 O + H + M ⇌ OH + M	$1.379 \times 10^{-42} T_g^{-1}$	[91]
R80 O + H ₂ ⇌ H + OH	$6.426 \times 10^{-26} T_g^{2.7} \exp(-3150/T_g)$	[91]
R81 O + HO ₂ ⇌ OH + O ₂	3.321×10^{-17}	[91]
R82 O + H ₂ O ₂ ⇌ OH + HO ₂	$1.599 \times 10^{-23} T_g^2 \exp(-2013/T_g)$	[91]
R143 O + O ₃ ⇌ O ₂ + O ₂	$2 \times 10^{-17} \cdot \exp(-2300.0/T_g)$	[6]
R144 O + O + O ₂ ⇌ O ₂ + O ₂	$2.45 \times 10^{-43} \cdot T_g^{-0.63}$	[6]
R145 O + O ₂ + O ₂ ⇌ O ₃ + O ₂	$8.62 \times 10^{-43} \cdot T_g^{-1.25}$	[6]
OH reactions		
R97 OH + H ₂ ⇌ H + H ₂ O	$3.587 \times 10^{-22} T_g^{1.51} \exp(-1726/T_g)$	[91]
R98 2OH + M ⇌ H ₂ O ₂ + M	$2.040 \times 10^{-46} T_g^{-0.37}$	[91]
R99 2OH ⇌ O + H ₂ O	$5.928 \times 10^{-26} T_g^{2.4} \exp(1062/T_g)$	[91]
R100 OH + HO ₂ ⇌ O ₂ + H ₂ O	$2.408 \times 10^{-17} \exp(252/T_g)$	[91]
R101 OH + H ₂ O ₂ ⇌ H ₂ O + HO ₂	$3.321 \times 10^{-18} \exp(-215/T_g)$	[91]
R102 OH + H ₂ O ₂ ⇌ H ₂ O + HO ₂	$2.823 \times 10^{-12} \exp(-14800/T_g)$	[91]
R105 OH + HO ₂ ⇌ O ₂ + H ₂ O	$8.303 \times 10^{-15} \exp(-8721/T_g)$	[91]
O(¹ D) reactions		
R11 O(¹ D) ↔ O	0.00563	[90]
R12 O(¹ D) ↔ O	2.11×10^{-05}	[90]
R114 O(¹ D) + O ₂ ⇒ O + O ₂ (b ¹ Σ)	$2.56 \times 10^{-17} \cdot \exp(+67/T_g)$	[33]
R115 O(¹ D) + O ₂ ⇒ O + O ₂	$7 \times 10^{-18} \cdot \exp(+67/T_g)$	[33]
R116 O(¹ D) + O ₃ ⇒ O + O + O ₂	2.3×10^{-16}	[6]
R117 O(¹ D) + O ₃ ⇒ O ₂ + O ₂	1.2×10^{-16}	[6]
R118 O(¹ D) + Ar ⇒ O + Ar	5.0×10^{-18}	[33]
R119 O(¹ D) + H ₂ ⇒ H + OH	1.2×10^{-16}	[6]

continuing on the next page...

... continued from the previous page

Reaction	Rate coefficient [m ³ /s or m ⁶ /s]	Refs.
O(¹ S) reactions		
R9 O(¹ S) ↔ O	0.000242	[90]
R10 O(¹ S) ↔ O(¹ D)	1.26	[90]
R120 O(¹ S) + O ₂ ⇒ O(¹ D) + O ₂	1.333 × 10 ⁻¹⁸ · exp(-850/T _g)	[6]
R121 O(¹ S) + O ₃ ⇒ O(¹ D) + O + O ₂	2.9 × 10 ⁻¹⁶	[6]
R122 O(¹ S) + O ₃ ⇒ O ₂ + O ₂	2.9 × 10 ⁻¹⁶	[6]
R123 O(¹ S) + O ₂ (a ¹ Δ) ⇒ O(¹ D) + O ₂ (b ¹ Σ)	3.6 × 10 ⁻¹⁷	[6]
R124 O(¹ S) + O ₂ (a ¹ Δ) ⇒ O + O + O	3.4 × 10 ⁻¹⁷	[6]
R125 O(¹ S) + O ₂ (a ¹ Δ) ⇒ O + O ₂ (a ¹ Δ)	1.3 × 10 ⁻¹⁶	[6]
R126 O(¹ S) + O ⇒ O(¹ D) + O	5 × 10 ⁻¹⁷ · exp(-301/T _g)	[6]
R127 O(¹ S) + H ₂ ⇒ O + H ₂	2.6 × 10 ⁻²²	[6]
O ₂ (a ¹ Δ) reactions		
R128 O ₂ (a ¹ Δ) + O ₂ ⇒ O ₂ + O ₂	6 × 10 ⁻²²	[33]
R129 O ₂ (a ¹ Δ) + O ₃ ⇒ O ₂ + O ₂ + O	5.2 × 10 ⁻¹⁷ · exp(-2840/T _g)	[33]
R130 O ₂ (a ¹ Δ) + O ⇒ O ₂ + O	7 × 10 ⁻²²	[33]
R131 O ₂ (a ¹ Δ) + Ar ⇒ O ₂ + Ar	6 × 10 ⁻²²	[33]
R132 O ₂ (a ¹ Δ) + H ₂ ⇒ OH + OH	2.8 × 10 ⁻¹⁵ · exp(-17906/T _g)	[6]
R133 O ₂ (a ¹ Δ) + H ₂ ⇒ O ₂ + H ₂	2.6 × 10 ⁻²⁵ · T _g ^{0.5}	[6]
R134 O ₂ (a ¹ Δ) + H ⇒ O + OH	1.3 × 10 ⁻¹⁷ · exp(-2530/T _g)	[6]
R135 O ₂ (a ¹ Δ) + H ⇒ O ₂ + H	5.2 × 10 ⁻¹⁷ · exp(-2530/T _g)	[6]
R136 O ₂ (a ¹ Δ) + HO ₂ ⇒ O ₂ + HO ₂	2 × 10 ⁻¹⁷	[6]
O ₂ (b ¹ Σ) reactions		
R137 O ₂ (b ¹ Σ) + O ₃ ⇒ O ₂ + O ₂ + O	1.8 × 10 ⁻¹⁷	[6]
R138 O ₂ (b ¹ Σ) + O ₂ ⇒ O ₂ (a ¹ Δ) + O ₂	4.3 × 10 ⁻²⁸ · T _g ^{2.4} · exp(-241/T _g)	[6]
R139 O ₂ (b ¹ Σ) + O ⇒ O ₂ (a ¹ Δ) + O	8 × 10 ⁻²⁰	[6]
R140 O ₂ (b ¹ Σ) + O ⇒ O ₂ + O(¹ D)	6 × 10 ⁻¹⁷ · T _g ^{-0.1} · exp(-4201/T _g)	[6]
R141 O ₂ (b ¹ Σ) + Ar ⇒ O ₂ + Ar	10 ⁻²²	[33]
R142 O ₂ (b ¹ Σ) + H ₂ ⇒ O ₂ (a ¹ Δ) + H ₂	10 ⁻¹⁸	[6]

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] I. V. Adamovich et al. “The 2017 Plasma Roadmap: Low temperature plasma science and technology”. In: *J. Phys. D: Appl. Phys.* 50.32 (2017), p. 323001.
- [2] T. Von Woedtke, H. R. Metelmann, and K. D. Weltmann. “Clinical plasma medicine: state and perspectives of in vivo application of cold atmospheric plasma”. In: *Contrib. Plasma Phys.* 54.2 (2014), pp. 104–117.
- [3] L. L. Alves et al. “Foundations of modelling of nonequilibrium low-temperature plasmas”. In: *Plasma Sources Sci. Technol.* 27.2 (2018), p. 023002.
- [4] Patrick J Roache. *Verification and validation in computational science and engineering*. Vol. 895. Hermosa Albuquerque, NM, 1998.
- [5] John S Carson. “Model verification and validation”. In: *Proceedings of the winter simulation conference*. Vol. 1. IEEE. 2002, pp. 52–58.
- [6] Igor V. Adamovich, Ting Li, and Walter R. Lempert. “Kinetic mechanism of molecular energy transfer and chemical reactions in low-temperature air-fuel plasmas”. In: *Philos. Trans. Royal Soc. A* 373.2048 (2015), p. 20140336.
- [7] Suo Yang et al. “Nanosecond pulsed plasma activated C₂H₄/O₂/Ar mixtures in a flow reactor”. In: *Journal of Propulsion and Power* (2016), pp. 1240–1252.
- [8] Mark J. Kushner. “A kinetic study of the plasma-etching process. I. A model for the etching of Si and SiO₂ in C_nF_m/H₂ and C_nF_m/O₂ plasmas”. In: *J. Appl. Phys.* 53.4 (1982), pp. 2923–2938.
- [9] Mounir Laroussi. “Low temperature plasma-based sterilization: overview and state-of-the-art”. In: *Plasma Process. Polym.* 2.5 (2005), pp. 391–400.
- [10] Mounir Laroussi. “Low-temperature plasmas for medicine?” In: *IEEE Trans. Plasma Sci.* 37.6 (2009), pp. 714–725.
- [11] Mark J. Kushner. “Hybrid modelling of low temperature plasmas for fundamental investigations and equipment design”. In: *J. Phys. D: Appl. Phys.* 42.19 (2009), p. 194013.
- [12] H. C. Kim et al. “Particle and fluid simulations of low-temperature plasma discharges: benchmarks and kinetic effects”. In: *J. Phys. D: Appl. Phys.* 38.19 (2005), R283.

- [13] J. Barnes and P. Hut. “A Hierarchical $O(N \log N)$ Force-Calculation Algorithm”. In: *Nature* 324 (1986), pp. 446–449. DOI: 10.1038/324446a0.
- [14] C. K. Birdsall and A. B. Langdon. *Plasma physics via computer simulation*. CRC Press, 2005.
- [15] R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. CRC Press, 1988.
- [16] John P. Verboncoeur. “Particle simulation of plasmas: review and advances”. In: *Plasma Phys. Control. Fusion*. 47.5A (2005), A231.
- [17] John P Verboncoeur, A Bruce Langdon, and NT Gladd. “An object-oriented electromagnetic PIC code”. In: *Computer Physics Communications* 87.1-2 (1995), pp. 199–211.
- [18] Carlo Benedetti et al. “ALaDyn: A High-Accuracy PIC Code for the Maxwell–Vlasov Equations”. In: *IEEE Transactions on plasma science* 36.4 (2008), pp. 1790–1798.
- [19] Lawrence Berkeley National Laboratory. *WarpX*. <https://ecp-warpX.github.io/>.
- [20] J-L Vay et al. “Modeling of a chain of three plasma accelerator stages with the WarpX electromagnetic PIC code on GPUs”. In: *Physics of Plasmas* 28.2 (2021), p. 023105.
- [21] Tech-X Corporation. *VSim*. <https://www.txcorp.com/vsim>.
- [22] COMSOL. *COMSOL Multiphysics Modeling Software*. <https://www.comsol.com>. Accessed: 2019-06-20. 2019.
- [23] Bruce Goplen et al. “User-configurable MAGIC for electromagnetic PIC calculations”. In: *Computer Physics Communications* 87.1-2 (1995), pp. 54–86.
- [24] Stellar Science. *ICEPIC*. <https://www.stellarscience.com/project/icepic/>. Accessed: 2021-08-09. 2021.
- [25] José A Bittencourt. *Fundamentals of plasma physics*. Springer Science & Business Media, 2013.
- [26] BD Dudson et al. “BOUT++: A framework for parallel plasma fluid simulations”. In: *Computer Physics Communications* 180.9 (2009), pp. 1467–1480.
- [27] TY Xia and XQ Xu. “Nonlinear fluid simulation of particle and heat fluxes during burst of ELMs on DIII-D with BOUT++ code”. In: *Nuclear Fusion* 55.11 (2015), p. 113030.

- [28] Abhishek Kumar Verma and Ayyaswamy Venkatraman. “SOMAFOAM: An Open-FOAM based solver for continuum simulations of low-temperature plasmas”. In: *Computer Physics Communications* 263 (2021), p. 107855.
- [29] Tech-X Corporation. *USim*. <https://www.txcorp.com/usim>.
- [30] ESI Group. *CFD-ACE+*. <https://www.esi-group.com/software-solutions/virtual-environment/cfd-multiphysics/ace-suite/cfd-ace>.
- [31] Plasma Matters and Eindhoven University of Technology. *PLASIMO*. <https://plasimo.phys.tue.nl/index.php>.
- [32] Seok Won Hwang, Ho-Jun Lee, and Hae June Lee. “Effect of electron Monte Carlo collisions on a hybrid simulation of a low-pressure capacitively coupled plasma”. In: *Plasma Sources Science and Technology* 23.6 (2014), p. 065040.
- [33] Annemie Bogaerts. “Effects of oxygen addition to argon glow discharges: a hybrid Monte Carlo-fluid modeling investigation”. In: *Spectrochimica Acta Part B: Atomic Spectroscopy* 64.11-12 (2009), pp. 1266–1279.
- [34] Matthew W Kunz, James M Stone, and Xue-Ning Bai. “Pegasus: a new hybrid-kinetic particle-in-cell code for astrophysical plasma dynamics”. In: *Journal of Computational Physics* 259 (2014), pp. 154–174.
- [35] K. Ding and M. A. Lieberman. “Reaction pathways for bio-active species in a He/H₂O atmospheric pressure capacitive discharge”. In: *J. Phys. D: Appl. Phys.* 48.3 (2014), p. 035401.
- [36] Zakari Eckert et al. “Kinetics of plasma-assisted oxidation of highly diluted hydrocarbon mixtures excited by a repetitive nanosecond pulse discharge”. In: *J. Phys. D: Appl. Phys.* 51.37 (2018), p. 374002.
- [37] Sang Ki Nam and John P. Verboncoeur. “Effect of electron energy distribution function on the global model for high power microwave breakdown at high pressures”. In: *Appl. Phys. Lett.* 92.23 (2008), p. 231502.
- [38] Sang Ki Nam and John P. Verboncoeur. “Effect of microwave frequency on breakdown and electron energy distribution function using a global model”. In: *Appl. Phys. Lett.* 93.15 (2008), p. 151504.
- [39] G. M. Parsey et al. *Non-equilibrium kinetics of a microwave-assisted jet flame: Global model and comparison with experiment*. The 40th IEEE International Conference on Plasma Science (ICOPS), San Francisco, California, USA, Jun 16-21. 2013. DOI: 10.1109/PLASMA.2013.6634762.

- [40] Yangyang Fu et al. “Investigation on the effect of nonlinear processes on similarity law in high-pressure argon discharges”. In: *Phys. Plasmas* 24.11 (2017), p. 113518.
- [41] Yangyang Fu et al. “Characterizing the dominant ions in low-temperature argon plasmas in the range of 1–800 Torr”. In: *Phys. Plasmas* 25.3 (2018), p. 033505. DOI: 10.1063/1.5020097.
- [42] Sang Ki Nam and John P. Verboncoeur. “Global model for high power microwave breakdown at high pressure in air”. In: *Comput. Phys. Commun.* 180.4 (Apr. 2009), pp. 628–635. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2008.12.013.
- [43] Rajesh Dorai, Khaled Hassouni, and Mark J. Kushner. “Interaction between soot particles and NO_x during dielectric barrier discharge plasma remediation of simulated diesel exhaust”. In: *J. Appl. Phys.* 88.10 (2000), pp. 6060–6071.
- [44] S. Pancheshnyi et al. *Computer code ZDPlasKin*. <https://www.zdplaskin.laplace.univ-tlse.fr>. University of Toulouse, LAPLACE, CNRS-UPS-INP, Toulouse, France. 2008.
- [45] A. H. Markosyan et al. “PumpKin: A tool to find principal pathways in plasma chemical models”. In: *Comput. Phys. Commun.* 185.10 (2014), pp. 2697–2702.
- [46] G. M. Parsey. “A global modeling framework for plasma kinetics: development and applications”. PhD thesis. Michigan State University, 2017.
- [47] G. M. Parsey, Y. Güçlü, and J. Krek. *Kinetic Global Model framework (KGMf)*. <http://ptsg.egr.msu.edu/KGMf/>.
- [48] L. C. Pitchford, S. V. O’Neil, and J. R. Rumble Jr. “Extended Boltzmann analysis of electron swarm experiments”. In: *Phys. Rev. A* 23.1 (1981), p. 294.
- [49] L. C. Pitchford et al. “Comparisons of sets of electron–neutral scattering cross sections and swarm parameters in noble gases: I. Argon”. In: *J. Phys. D: Appl. Phys.* 46.33 (2013), p. 334001.
- [50] L. L. Alves et al. “Comparisons of sets of electron–neutral scattering cross sections and swarm parameters in noble gases: II. Helium and neon”. In: *J. Phys. D: Appl. Phys.* 46.33 (2013), p. 334002.
- [51] M. C. Bordage et al. “Comparisons of sets of electron–neutral scattering cross sections and swarm parameters in noble gases: III. Krypton and xenon”. In: *J. Phys. D: Appl. Phys.* 46.33 (2013), p. 334003.

- [52] J Bretagne et al. “Relativistic electron-beam-produced plasmas. I. Collision cross sections and loss function in argon”. In: *J. Phys. D: Appl. Phys.* 19.5 (1986), p. 761.
- [53] Peng-Cheng Zhao et al. “Validity of the two-term Boltzmann approximation employed in the fluid model for high-power microwave breakdown in gas”. In: *Chin. Phys. B* 23.5 (2014), p. 055101.
- [54] Tz. B. Petrova, H. D. Ladouceur, and A. P. Baronavski. “Nonequilibrium dynamics of laser-generated plasma channels”. In: *Phys. Plasmas* 15.5 (2008), p. 053501.
- [55] Bowen Sun et al. “Global model of cold atmospheric He + air plasmas: A comparison of Maxwellian and non-Maxwellian EEDFs”. In: *Phys. Plasmas* 26.12 (2019), p. 123508. DOI: 10.1063/1.5124023. eprint: <https://doi.org/10.1063/1.5124023>. URL: <https://doi.org/10.1063/1.5124023>.
- [56] J. T. Gudmundsson. “On the effect of the electron energy distribution on the plasma parameters of an argon discharge: a global (volume-averaged) model study”. In: *Plasma Sources Sci. Technol.* 10.1 (2001), p. 76.
- [57] V. Vahedi and M. Surendra. “A Monte Carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges”. In: *Comput. Phys. Commun.* 87.1-2 (1995), pp. 179–198.
- [58] Mark J. Kushner. “Monte-Carlo simulation of electron properties in rf parallel plate capacitively coupled discharges”. In: *J. Appl. Phys.* 54.9 (1983), pp. 4958–4965.
- [59] S. F. Biagi. “Monte Carlo simulation of electron drift and diffusion in counting gases under the influence of electric and magnetic fields”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 421.1-2 (1999), pp. 234–240.
- [60] Michele Renda and Dan Andrei Ciubotaru. “Betaboltz: a Monte-Carlo simulation tool for gas scattering processes”. In: *arXiv preprint arXiv:1901.08140* (2019).
- [61] M. Rabie and C. M. Franck. “METHES: A Monte Carlo collision code for the simulation of electron transport in low temperature plasmas”. In: *Comput. Phys. Commun.* 203 (June 2016), pp. 268–277. DOI: 10.1016/j.cpc.2016.02.022.
- [62] W. L. Morgan and B. M. Penetrante. “ELENDIF: A time-dependent Boltzmann solver for partially ionized plasmas”. In: *Comput. Phys. Commun.* 58.1-2 (1990), pp. 127–152.
- [63] S. F. Biagi. *Magboltz-transport of electrons in gas mixtures*. <http://magboltz.web.cern.ch/magboltz/>. 2000.

- [64] G. J. M. Hagelaar. *BOLSIG+*. <https://www.bolsig.laplace.univ-tlse.fr>. Accessed: 2018-08-31. Dec. 2018.
- [65] A. Luque. *BOLOS*. <https://pypi.python.org/pypi/bolos>. Accessed: 2017-03-31. 2014.
- [66] A. Tejero-del Caz et al. “The LisOn KInetics Boltzmann solver”. In: *Plasma Sources Sci. Technol.* 28.4 (2019), p. 043001.
- [67] J. Stephens. “A multi-term Boltzmann equation benchmark of electron-argon cross-sections for use in low temperature plasma models”. In: *J. Phys. D: Appl. Phys.* 51.12 (2018), p. 125203.
- [68] S. L. Lin, R. E. Robson, and E. A. Mason. “Moment theory of electron drift and diffusion in neutral gases in an electrostatic field”. In: *J. Chem. Phys.* 71.8 (1979), pp. 3483–3498.
- [69] R. E. Robson and K. F. Ness. “Velocity distribution function and transport coefficients of electron swarms in gases: spherical-harmonics decomposition of Boltzmann’s equation”. In: *Phys. Rev. A* 33.3 (1986), p. 2068.
- [70] Ronald Douglas White et al. “Is the classical two-term approximation of electron kinetic theory satisfactory for swarms and plasmas?” In: *J. Phys. D: Appl. Phys.* 36.24 (2003), p. 3125.
- [71] H. C. Kim and J. P. Verboncoeur. “Transition of window breakdown from vacuum multipactor discharge to rf plasma”. In: *Phys. Plasmas* 13.12 (2006), p. 123506. DOI: <https://doi.org/10.1063/1.2403782>.
- [72] Peng Zhao and SH Lam. “Toward computational singular perturbation (CSP) without eigen-decomposition”. In: *Combustion and Flame* 209 (2019), pp. 63–73.
- [73] J. Krek, Y. Fu, and J. P. Verboncoeur. *Benchmarking the Kinetic Global Model framework (KGMf): EEDF evaluations in low-temperature argon plasmas*. The 46th Pulsed Power & Plasma Science (PPPS), Orlando, Florida, USA, June 22-28. 2019.
- [74] Ronald L Iman and WJ Conover. “Small sample sensitivity analysis techniques for computer models. with an application to risk assessment”. In: *Communications in statistics-theory and methods* 9.17 (1980), pp. 1749–1842.
- [75] Max D Morris. “Factorial sampling plans for preliminary computational experiments”. In: *Technometrics* 33.2 (1991), pp. 161–174.

- [76] Andrea Saltelli. “Making best use of model evaluations to compute sensitivity indices”. In: *Comput. Phys. Commun.* 145.2 (2002), pp. 280–297.
- [77] Andrea Saltelli et al. *Sensitivity analysis in practice: a guide to assessing scientific models*. Vol. 1. Wiley Online Library, 2004.
- [78] Andrea Saltelli et al. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.
- [79] Miles M Turner. “Uncertainty and error in complex plasma chemistry models”. In: *Plasma Sources Science and Technology* 24.3 (2015), p. 035027.
- [80] Miles M Turner. “Uncertainty and sensitivity analysis in complex plasma chemistry models”. In: *Plasma Sources Science and Technology* 25.1 (2016), p. 015003.
- [81] Peter Koelman et al. “A Comprehensive Chemical Model for the Splitting of CO₂ in Non-Equilibrium Plasmas”. In: *Plasma Processes and Polymers* 14.4-5 (2017), p. 1600155.
- [82] Tianfeng Lu and Chung K Law. “Linear time reduction of large kinetic mechanisms with directed relation graph: n-Heptane and iso-octane”. In: *Combustion and flame* 144.1-2 (2006), pp. 24–36.
- [83] Ralph Lehmann. “Determination of dominant pathways in chemical reaction systems: An algorithm and its application to stratospheric chemistry”. In: *Journal of atmospheric chemistry* 41.3 (2002), pp. 297–314.
- [84] Ralph Lehmann. “An algorithm for the determination of all significant pathways in chemical reaction systems”. In: *Journal of atmospheric chemistry* 47.1 (2004), pp. 45–78.
- [85] K. Ding, M. A. Lieberman, and A. J. Lichtenberg. “Hybrid model of neutral diffusion, sheaths, and the α to γ transition in an atmospheric pressure He/H₂O bounded rf discharge”. In: *Journal of Physics D: Applied Physics* 47.30 (2014), p. 305203.
- [86] Martin Hanicinec, Sebastian Mohr, and Jonathan Tennyson. “Fast species ranking for iterative species-oriented skeletal reduction of chemistry sets”. In: *Plasma Sources Science and Technology* 29.12 (2020), p. 125024.
- [87] Tianfeng Lu and Chung K Law. “A directed relation graph method for mechanism reduction”. In: *Proceedings of the Combustion Institute* 30.1 (2005), pp. 1333–1341.
- [88] Tianfeng Lu and Chung K Law. “On the applicability of directed relation graphs to the reduction of reaction mechanisms”. In: *Combustion and Flame* 146.3 (2006), pp. 472–483.

- [89] Peter Koelman et al. “Uncertainty analysis with a reduced set of input uncertainties selected using pathway analysis”. In: *Plasma Sources Science and Technology* 28.7 (2019), p. 075009.
- [90] LXCat. *Plasma Data Exchange Project*. <http://www.lxcat.net/>. Accessed: 2019-06-20. 2019.
- [91] Gregory P. Smith et al. *GRI-Mech 3.0*. URL: http://www.me.berkeley.edu/gri_mech/.
- [92] Janez Krek et al. “Benchmark of the KGMf with a coupled Boltzmann equation solver”. In: *Comput. Phys. Commun.* 260 (2021), p. 107748. DOI: 10.1016/j.cpc.2020.107748.
- [93] ANSYS. *ChemKin*. <https://www.ansys.com/products/fluids/ansys-chemkin-pro>.
- [94] Bowman C.T. et al. *GRI-Mech 2.11*. http://www.me.berkeley.edu/gri_mech/. 2014.
- [95] Y. P. Raizer, V. I. Kisin, and J. E. Allen. *Gas Discharge Physics*. Springer Berlin Heidelberg, 2011. ISBN: 9783642647604.
- [96] G. J. M. Hagelaar and L. C. Pitchford. “Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models”. In: *Plasma Sources Sci. Technol.* 14.4 (2005), p. 722.
- [97] A. V. Phelps and L. C. Pitchford. “Anisotropic scattering of electrons by N₂ and its effect on electron transport”. In: *Phys. Rev. A* 31.5 (1985), p. 2932.
- [98] Michael A. Lieberman and Alan J. Lichtenberg. *Principles of plasma discharges and materials processing*. John Wiley & Sons, 2005.
- [99] Biagi database, v8.9. <http://www.lxcat.net/>. retrieved: July 2012.
- [100] Lutz Baars-Hibbe et al. “Micro-Structured Electrode Arrays: Characterization of High Frequency Discharges at Atmospheric Pressure”. In: *Plasma Process Polym* 2.3 (2005), p. 174. DOI: 10.1002/ppap.200400043.
- [101] Yangyang Fu, Peng Zhang, and John P. Verboncoeur. “Gas breakdown in atmospheric pressure microgaps with a surface protrusion on the cathode”. In: *Appl. Phys. Lett.* 112.25 (2018), p. 254102. DOI: 10.1063/1.5037688.

- [102] Yangyang Fu et al. “Gas breakdown and its scaling law in microgaps with multiple concentric cathode protrusions”. In: *Appl. Phys. Lett.* 114.1 (2019), p. 014102.
- [103] Y. Y. Lau, J. P. Verboncoeur, and H. C. Kim. “Scaling laws for dielectric window breakdown in vacuum and collisional regimes”. In: *Appl. Phys. Lett.* 89.26 (2006), p. 261501.
- [104] Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*. Vol. 12. Siam, 2013.
- [105] Roger Ghanem, David Higdon, and Houman Owhadi. *Handbook of uncertainty quantification*. Vol. 6. Springer, 2017.
- [106] Timothy John Sullivan. *Introduction to uncertainty quantification*. Vol. 63. Springer, 2015.
- [107] Andrea Saltelli et al. “Sensitivity analysis practices: Strategies for model-based inference”. In: *Reliability Engineering & System Safety* 91.10-11 (2006), pp. 1109–1125.
- [108] Yukikazu Itikawa. “Cross sections for electron collisions with oxygen molecules”. In: *Journal of Physical and Chemical Reference Data* 38.1 (2009), pp. 1–20.
- [109] Jung-Sik Yoon et al. “Cross sections for electron collisions with hydrogen molecules”. In: *Journal of Physical and Chemical Reference Data* 37.2 (2008), pp. 913–931.
- [110] C. Winters et al. “Measurements and kinetic modeling of atomic species in fuel-oxidizer mixtures excited by a repetitive nanosecond pulse discharge”. In: *J. Phys. D: Appl. Phys.* 51.1 (2018), p. 015202.
- [111] Bruce L Clarke. “Complete set of steady states for the general stoichiometric dynamical system”. In: *The Journal of Chemical Physics* 75.10 (1981), pp. 4970–4979.
- [112] Christophe H Schilling, David Letscher, and Bernhard Ø Palsson. “Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective”. In: *Journal of theoretical biology* 203.3 (2000), pp. 229–248.
- [113] Aurélie Bellemans, Nicholas Deak, and Fabrizio Bisetti. “Development of skeletal kinetics mechanisms for plasma-assisted combustion via principal component analysis”. In: *Plasma Sources Science and Technology* 29.2 (2020), p. 025020.
- [114] Tibor Nagy and Tamás Turányi. “Reduction of very large reaction mechanisms using methods based on simulation error minimization”. In: *Combustion and Flame* 156.2 (2009), pp. 417–428.

- [115] Perrine Pepiot-Desjardins and Heinz Pitsch. “An efficient error-propagation-based reduction method for large chemical kinetic mechanisms”. In: *Combustion and Flame* 154.1-2 (2008), pp. 67–81.
- [116] Aurélie Bellemans et al. “P-DRGEP: a novel methodology for the reduction of kinetics mechanisms for plasma-assisted combustion applications”. In: *Proceedings of the Combustion Institute* (2020).
- [117] Sumio Ashida, C Lee, and MA Lieberman. “Spatially averaged (global) model of time modulated high density argon plasmas”. In: *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* 13.5 (1995), pp. 2498–2507.
- [118] P Dohnal et al. “Recombination in He/Ar afterglow plasma at low temperatures”. In: *Proceedings of the 21st Annual Conference of Doctoral Students, Charles University, Prague*. 2012, p. 18.
- [119] S. Pancheshnyi et al. “The LXCat project: Electron scattering cross sections and swarm parameters for low temperature plasma modeling”. In: *Chem. Phys.* 398 (2012), pp. 148–153.
- [120] Annemie Bogaerts and Renaat Gijbels. “Effects of adding hydrogen to an argon glow discharge: overview of relevant processes and some qualitative explanations”. In: *Journal of Analytical Atomic Spectrometry* 15.4 (2000), pp. 441–449.
- [121] JE Velazco, JH Kolts, and DW Setser. “Rate constants and quenching mechanisms for the metastable states of argon, krypton, and xenon”. In: *The Journal of Chemical Physics* 69.10 (1978), pp. 4357–4373.