Computer Physics Communications 183 (2012) 2550–2560

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A parallel hybrid numerical algorithm for simulating gas flow and gas discharge of an atmospheric-pressure plasma jet

K.-M. Lin [a], M.-H. Hu [a], C.-T. Hung [a], J.-S. Wu [a,*], F.-N. Hwang [b], Y.-S. Chen [c], G. Cheng [d]

[a] Department of Mechanical Engineering, National Chiao Tung University, Hsinchu, Taiwan
[b] Department of Mathematics, National Central University, Taoyuan, Taiwan
[c] National Space Organization, National Applied Research Laboratory, Hsinchu, Taiwan
[d] Department of Mechanical Engineering, University of Alabama, Birmingham, AL, USA

ARTICLE INFO

Article history:
Received 29 April 2012
Received in revised form
2 July 2012
Accepted 10 July 2012
Available online 20 July 2012

Keywords:
Atmospheric-pressure plasma jet (APPJ)
Navier–Stokes (N–S) equation solver
Plasma fluid model (PFM)
Temporal multi-scale method (TMSM)

ABSTRACT

Development of a hybrid numerical algorithm which couples weakly with the gas flow model (GFM) and the plasma fluid model (PFM) for simulating an atmospheric-pressure plasma jet (APPJ) and its acceleration by two approaches is presented. The weak coupling between gas flow and discharge is introduced by transferring between the results obtained from the steady-state solution of the GFM and cycle-averaged solution of the PFM respectively. Approaches of reducing the overall runtime include parallel computing of the GFM and the PFM solvers, and employing a temporal multi-scale method (TMSM) for PFM. Parallel computing of both solvers is realized using the domain decomposition method with the message passing interface (MPI) on distributed-memory machines. The TMSM considers only chemical reactions by ignoring the transport terms when integrating temporally the continuity equations of heavy species at each time step, and then the transport terms are restored only at an interval of time marching steps. The total reduction of runtime is 47% by applying the TMSM to the APPJ example presented in this study. Application of the proposed hybrid algorithm is demonstrated by simulating a parallel-plate helium APPJ impinging onto a substrate, which the cycle-averaged properties of the 200th cycle are presented. The distribution patterns of species densities are strongly correlated by the background gas flow pattern, which shows that consideration of gas flow in APPJ simulations is critical.

1. Introduction

Low-temperature atmospheric-pressure plasma jet (APPJ hereafter) has attracted much attention recently because they do not require the use of expensive vacuum equipment. The applications of the APPJ include surface modification, surface cleaning, thin-film deposition, sterilization, and tooth bleaching, to name a few [1–5]. The design of these plasma devices mainly depends on time-consuming and expensive trial-and-error approaches, which is economically inefficient. It is limited and difficult, if possible, to comprehend completely the complex physics and chemistry associated with the APPJ through the experimental methods. Therefore, numerical simulation provides an alternative approach in revealing the complex physics and chemistry of APPJs.

Two major approaches, Particle-in-Cell with Monte-Carlo collision (PIC/MCC) and plasma fluid modeling (PFM hereafter), have been widely used for low-temperature plasma simulations. Although the PIC/MCC approach solves the Boltzmann equation statistically, it is very time-consuming for the high-pressure condition because a large amount of pseudo particles is needed for obtaining an accurate solution. PFM, which assumes the plasma as a continuum, is often employed to model gas discharges and requires less computational time than the PIC/MCC approach, should the pressure be not too low. Nevertheless, applications accompanied with a large-scale (computational) domain and/or complex chemistry (species and reactions), which is not uncommon in practice, could lead to unacceptable computational time even using PFM. Acceleration of the fluid modeling is thus strongly required in simulating realistic gas discharges.

Gas discharge simulation considering diffusion without convection for heavy species (i.e., ignoring fluid dynamics effect) is considered to be valid in the low-pressure condition. However, fluid dynamics is expected to have a strong impact on the gas discharge in the high-pressure condition such as the APPJ. Thus, it is necessary to properly model and integrate the gas flow and gas discharge simultaneously for a better understanding the APPJ. However, it is found that only very few studies in the literature have focused on this subject [6–9]. This is mainly because that a complete simulation of the APPJ coupling the fluid dynamics and gas discharge often

* Corresponding author. Tel.: +886 3 573 1693; fax: +886 3 611 0023.
E-mail address: chongsin@faculty.nctu.edu.tw (J.-S. Wu).

0010-4655/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2012.07.004

takes from weeks up to months of runtime. Specifically speaking, simulation of gas discharge often takes generally about 90% of the overall runtime for the simulation of the APPJ because of a very small time step limited by the very light electron. In other words, the bottleneck for speeding up the APPJ simulation is to shorten the runtime consumed by the modeling of gas discharge.

There have been almost no reports on parallel computing of PFM found in the literature, apart from Wu's group [10–12]. The parallel fully implicit Newton–Krylov–Schwarz (NKS) algorithm was employed to solve the PFM [10,11]. Although the speedup of parallel computing for the problem solved with fully implicit NKS algorithm is scalable up to hundreds of processors, the overall computational time is too large for realistic APPJ simulations. The numerical difficulty can be resolved by using the so-called semi-implicit method with parallel computing, which solves the governing equation of fluid modeling sequentially with proper linearization of the source terms of the Poisson equation and the electron energy density equation [12].

For the PFM employed in this study, the evolution of gas discharge is modeled by the Poisson equation, species continuity equations, and the electron energy density equation in this study. It is known that electrons respond extremely fast to the temporal variation of the electric field, leading to very large transport properties (i.e., mobility and diffusivity), whereas ions respond relatively slow to the temporal variation of the electric field. The neutrals transported by diffusion are even slower if compared with the drift of charged species induced by the electric field. The time step size used in solving the PFM is generally constrained by the electron motion, in the order of 10^{-10} s, and must be small enough to resolve the dynamics of electrons. This leads to the possibility of neglecting the transport of heavy species as compared to chemical reaction at each time step without losing much numerical accuracy of the simulation. Realization of this concept can further shorten the runtime for solving the PFM by using the temporal multi-scale method (TMSM), as presented later in this study.

Therefore, the objective of this study is to develop an efficient hybrid numerical algorithm to couple the gas flow model (GFM) and PFM utilizing parallel computing, which can be used for the practical simulation of APPJ. Two approaches to reduce the runtime will be explored. These include parallel computing of gas flow and gas discharge solvers using domain decomposition with MPI on distributed-memory machines and TMSM for the PFM. The simulation of realistic helium APPJ impinging onto a substrate is employed to demonstrate the developed numerical approach/solvers, and its computational efficiency.

The remainder of the paper is organized as follows. Section 2 describes the numerical methods developed in this study, followed by the results presented in Section 3. Finally, the major findings of the present study are summarized at the end of this paper.

2. Description of the model

The simulation of APPJ consists of two parts, the GFM and the PFM. The GFM considers the background gas flow field including the momentum exchange with heavy charged species and gas heating produced by energy transfer due to electron-neutral elastic collisions and Joule heating of ions. The PFM considers the plasma physics and chemistry accompanied with the convection of the background gas flow field. Theoretically, modeling the APPJ requires a strong coupling of the GFM and the PFM at each time step. For example, it is necessary to adopt the strong coupling of the GFM and the PFM to study the transition from non-thermal to thermal plasma with high interaction between gas flow and discharge as reported [9]. However, the characteristic time scales of the GFM and the PFM are 1.0–10 μs and 0.01–1.0 ns respectively, in which very large timescale differences (3–6 orders of magnitude) exist. Strong coupling at each time step of the GFM and the PFM becomes unnecessary and unrealistic for a steady background gas flow; hence, these two models are integrated with a weakly coupling algorithm. Each model is described first in turn, and then followed by the coupling algorithm.

2.1. The gas flow model (GFM) solver

The GFM employed in the present study is a two-dimensional planar and axisymmetric flow solver developed in our group [13], which simulates the background gas flow as a continuum by solving a set of governing equations, including the continuity, Navier–Stokes (N–S), energy, species transport equations, and the equation of state for ideal gases. The general form of two-dimensional planar governing equations is written in the Cartesian tensor as

$$\frac{\partial(\rho\varphi)}{\partial t}+\frac{\partial}{\partial X_i}(\rho V_i\varphi)=\frac{\partial}{\partial X_i}\left(\mu_\varphi\frac{\partial\varphi}{\partial X_i}\right)+S_\varphi \tag{1}$$

where t is the time, $X_i=(x,y)$ is the position vector, $V_i=(u,v)$ is the velocity vector, μ_φ is an effective diffusion coefficient, S_φ is the source term, ρ is the fluid density, and $\varphi=(1,u,v,h_t,Y_i)$ represents for the variables for the mass, momentum, energy, and mass fraction of the ith species, respectively. h_t $\left(h_t=\sum_i Y_i h_i+\frac{1}{2}\sum_{j=1}^{2}V_j^2\right.$ and $\left.h_i=\int C_{p,i}dT\right)$ is the total enthalpy, where $C_{p,i}$ is the specific heat capacity of the ith species at constant pressure and T is the mixture temperature of the background gas flow.

The temperature distribution of solids (e.g., electrode and dielectric material) is obtained by solving the steady state heat conduction equation since we are simulating the experiment when the solids are in their steady states. The steady-state heat conduction equation is written as

$$\frac{\partial}{\partial X_i}\left(k\frac{\partial T}{\partial X_i}\right)+S_{th,solid}=0 \tag{2}$$

where k is the thermal conductivity, $S_{th,solid}$ is the heat source/sink term of the solid material. Conjugate heat transfer is considered by applying the heat flux continuity at the gas–solid interfaces.

The governing equations of the GFM are solved using a cell-centered finite-volume method, with an extended SIMPLE (Semi-Implicit Method for Pressure Linked Equations) scheme [13]. A second-order upwind scheme with linear reconstruction is used to evaluate the inviscid flux across the cell interface. A flux limiter is used to prevent the occurrence of local extrema from being introduced by the data reconstruction. Pressure smoothing is employed to avoid the pressure oscillations on a collocation grid. The use of the above numerical approaches allows the GFM solver to simulate both compressible and incompressible flows with a wide range of speeds. The computation performed by the flow solver is also parallelized using the domain decomposition approach. Detailed numerical implementation and validations of the GFM solver can be found in [13].

2.2. The plasma fluid model (PFM) solver

The PFM was developed to include the plasma physics and chemistry considering the effect of background gas flow. The governing equations for the PFM are similar to those solved by Lin et al. [12] and are briefly described below for completeness. The continuity equation for a general species is written as

$$\frac{\partial n_{e,i,uc}}{\partial t}+\vec{\nabla}\cdot\vec{\Gamma}_{e,i,uc}=S_{e,i,uc} \tag{3}$$

where subscripts e, i, and uc represents the electron, ions, and uncharged neutral species respectively. The variable n represents the number density, S is the source related to species chemistry, and $\vec{\Gamma}$ is the species flux vector expressed based on the drift-diffusion approximation including the convection effect due to the background gas flow for heavy species as

$$\vec{\Gamma}_e = -\mu_e n_e \vec{E} - D_e \vec{\nabla} n_e \tag{4}$$

$$\vec{\Gamma}_i = \text{sign}(q_i)\mu_i n_i \vec{E} - D_i \vec{\nabla} n_i + \vec{u} n_i \tag{5}$$

$$\vec{\Gamma}_{uc} = -D_{uc} \vec{\nabla} n_{uc} + \vec{u} n_{uc} \tag{6}$$

where q, $\vec{E}$, $\vec{u}$, μ, and D are the species charge, the electric field, the flow velocity vector, the mobility, and the diffusivity, respectively. It is noted that the electron flux does not include the convection of the background gas flow since the speed of the electron induced by the drift of the electric field is much faster than that transported by the convection of the background gas flow. The electric field ($\vec{E} = -\nabla\phi$) is obtained by solving the Poisson equation

$$\nabla \cdot (\varepsilon \nabla \phi) = -\sum_{j=1}^{N_c} q_j n_j \tag{7}$$

where ε is the local permittivity, which is a function of space (i.e., material properties of the media), and N_c is the total number of charged species including electron and ions.

The electron temperature (T_e) is obtained by solving the electron energy density equation written as

$$\frac{\partial (n_\varepsilon)}{\partial t} + \vec{\nabla} \cdot \vec{\Gamma}_{n_\varepsilon} = S_{n_\varepsilon} \tag{8}$$

where $n_\varepsilon \left(= \frac{3}{2} n_e k_B T_e\right)$ is the electron energy density, T_e is the electron temperature, and k_B is the Boltzmann constant. S_{n_ε} represents the source term of the energy density equation including the energy loss due to elastic collision, ionization, excitation, dissociation of electron-related chemistry, and the energy gain because of the Joule heating from the electric field. There is no energy equation solved for ions and various neutral species since they are assumed to be in thermal equilibrium with the background gas. The $\vec{\Gamma}_{n_\varepsilon}$ is the corresponding electron energy density flux and is written as

$$\vec{\Gamma}_{n_\varepsilon} = -\frac{5}{3}\mu_e \vec{E} n_\varepsilon - \frac{5}{3} D_e \vec{\nabla} n_\varepsilon. \tag{9}$$

The governing equations of PFM are discretized by using the backward Euler's method in the temporal domain and the collocated cell-centered finite volume method with the Scharfetter-Gummel scheme for the fluxes of continuity equations and the electron energy density equation on the spatial domain. The numerical computation of the PFM is also parallelized with domain decomposition. Detailed implementation of the PFM can be found in [12].

2.3. Hybrid algorithm for coupling the GFM and PFM solvers

Challenge in coupling the GFM and the PFM solvers lies in the fact that the large timescale difference of characteristic time between the background gas flow and the gas discharge. As mentioned earlier, strong coupling at each time step of discharge simulation is very time consuming and unnecessary if transient gas flow field is not important. Thus, we adopt a weakly coupling algorithm in this study. Before describing the details of the coupling algorithm, we introduce the source terms of the momentum and energy equations, which appear in the GFM solver, due to gas discharge.

The cycle-averaged momentum source term considered in this study involves the electrohydrodynamic force [14] and can be written as

$$S_m = \frac{1}{t_p}\int_0^{t_p} \left(\sum_i q_i n_i \vec{E} - e n_e \vec{E} - \sum_i k_B T_g \vec{\nabla} n_i - k_B T_e \vec{\nabla} n_e \right) dt \tag{10}$$

where k_B is the Boltzmann constant and t_p is the period of the driving power source for the discharge. The elastic collision between the electron and background gas, and the summation of Joule heating of ions are integrated and averaged for each cycle and outputted as the source term of the energy equation of the GFM. The cycle-averaged energy source can be written as

$$S_{h_t} = \frac{1}{t_p}\int_0^{t_p} \left(\sum_i q_i \Gamma_i \vec{E} + 3\frac{m_e}{M} n_e k_B \nu_m \left(T_e - T_g\right) \right) dt \tag{11}$$

where ν_m is the momentum exchange collision frequency between the electron (mass m_e) and the background gas (mass M), and T_g is the background gas mixture temperature.

Fig. 1 shows the concept of a weakly coupling algorithm of the PFM and GFM solvers, and Fig. 2 shows the flowchart of a complete APPJ simulation. The coupling starts from the PFM solver, and then passes cycle-averaged simulation data, such as the momentum and energy source terms, to the GFM solver. The cycle-averaged momentum and energy source terms generated from the PFM solver are passed to the GFM solver for every two discharge cycles (for the discharge driven by kHz level of power source). Next, the GFM solver takes the momentum and energy source term provided by the PFM solver and solves the governing equations to obtain the steady-state flow velocity, temperature, and number densities of the background gases. These flow properties are provided as the input for PFM solver to evaluate the convective flux, and the local plasma chemistry partly determined by the temperature and number densities of the background gases. The source terms calculated by the PFM solver and the flow properties obtained by the GFM solver are transferred back and forth to account for the effect of discharge-gas flow interactions. This process may take several iterations between the GFM and the PFM solvers to obtain a converged solution for the GFM solver. In the current study, it is observed that three iterations between the GFM solver and the PFM solvers are sufficient to obtain a converged solution for the GFM solver. For the GFM solver, the converged solution represents the velocities, temperature, and the densities of background gases reach the steady level. The physical properties of gas flow obtained by the GFM solver do not change much after several iterations with the PFM once the converged solution is obtained.

It is noted that the PFM requires more time steps to achieve the physically quasi-steady state for the properties such as species number densities near the substrate surface even though the GFM solver has already reached the converged solution as shown in Fig. 2. The species generated from the discharge are advected by the flow field obtained from the GFM solver.

2.4. Acceleration of the hybrid coupling algorithm

We have applied two numerical approaches in accelerating the hybrid algorithm for coupling the gas flow and gas discharge solvers. They include parallel computing of both solvers and a multi-scale method for the PFM.

Parallel computing of GFM and PFM solvers

We have applied parallel computing for the GFM and PFM solvers. It is based on a domain decomposition method with

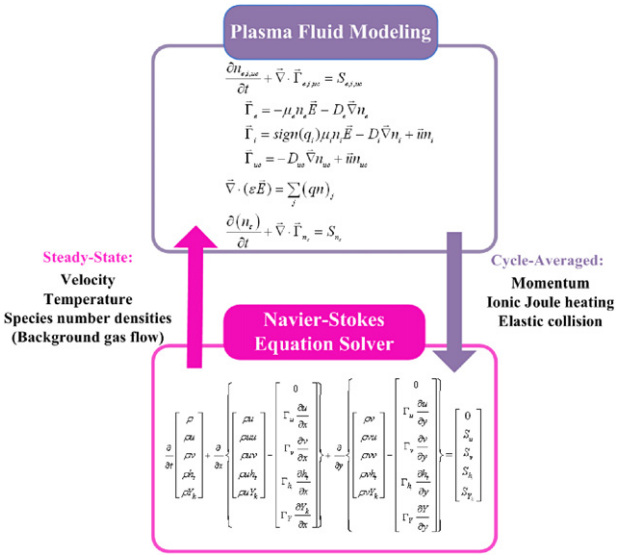

Fig. 1. The coupling of the GFM and the PFM solvers.

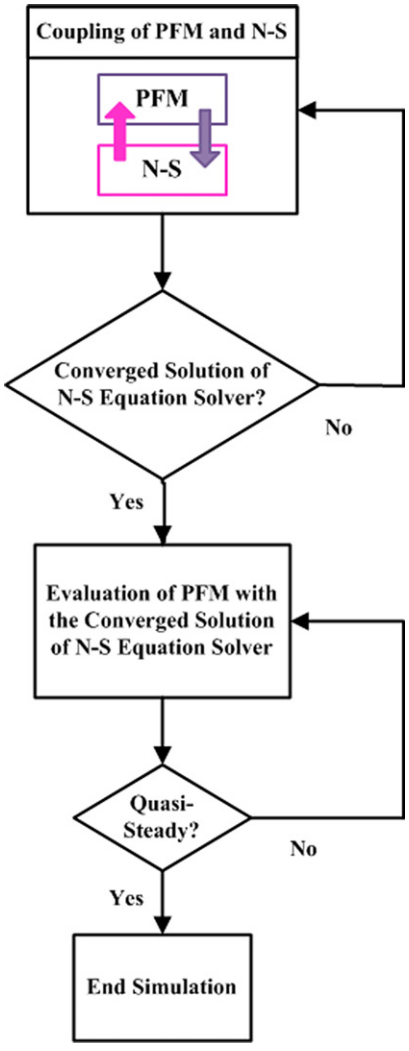

Fig. 2. The flowchart of complete APPJ simulation.

MPI protocol and implemented on distributed-memory machines. Both codes employed the GMRES [15] matrix solver with preconditioning using the additive Swartz method (ASM) [16]. Details and validation of the implementation can be found in [13,12], respectively, and are not repeated here for brevity.

Temporal multi-scale method (TMSM) for the PFM

To capture the electron dynamics that determines plasma physics and chemistry, the simulation time step is constrained by the characteristic time scale of electron transport. As mentioned earlier, there is no need to consider the transport of heavy species, in addition to chemical reactions, during each electron time step. This motivates us to develop the temporal multi-scale method (TMSM) for the PFM solver to further reduce runtime, as presented in the following.

The idea of TMSM is to ignore the transport terms of the continuity equations for heavy species when integrating them in time and restore the transport terms only at every certain (large) number of time steps. This simplifies the numerical solution of solving the continuity equations for heavy species and substantially reduces the computational time, since there is no need to solve the matrix equation during most time steps, which is very time-consuming.

Fig. 3 illustrates the TMSM idea in detail. It shows five typical equations (No. 1, 2, 4, 6, 7) that need to be solved at each time step in the original PFM along with those simplified equations (No. 3, 5) for ions and neutral species. Time scale factors based on the electron time step are termed as "tf_i" and "tf_{uc}" for ions and neutral (uncharged) species respectively. For example, "$tf_i = 5$" represents that, for every 5 time steps, a complete form of the continuity equation has to be solved for all ion species. In Fig. 3, the numbers above the ticks of each horizontal line represent the corresponding equations solved at certain specified time levels. The PFM is solved for all the equations in its original sequence at each time step. The Poisson equation, the electron continuity equation, and the electron energy density equation are always solved in their complete forms. For other heavy species (ions and neutrals), their continuity equations are solved in simplified forms by only considering the source terms at most of time steps. In doing so, there is no need to solve these simplified equations with matrix solvers, which is time-consuming. Only at a preset time interval, controlled by the time scale factors (e.g., $tf_i = 5$ and $tf_{uc} = 10$ in Fig. 3), are the ion and neutral continuity equations then solved in the complete forms to restore the transport effects. In practice, these time scale factors can be as high as 1,000 found in the current study. Results of runtime reduction and the accuracy of the numerical solution using the TMSM will be presented later. Some details of the calculation procedure for restoring the transport effect are introduced next.

Each time when the time interval equals an integral times of tf_i or tf_{uc} (i.e., $\Delta t_{i,uc} = tf_{i,uc} \times \Delta t$), the discretized continuity equations of ions and neutrals can be written as

$$\left(n_{i,uc}^{k+1} - n_{i.uc}^{k}\right) + \Delta t_{i,uc} \times \vec{\nabla} \cdot \vec{\Gamma}_{i,uc} = \Delta t \times S_{i,uc}^{k} \tag{12}$$

where the superscripts k and $k+1$ represent the previous and current time level respectively. It is noted that the time step size of the source term is different from that of the transport term since the source term is evaluated at every time step. The flux terms of neutrals are the same as Eq. (6), while those of ions can be written as

$$\vec{\Gamma}_i = \text{sign}(q_i)\mu_i n_i \vec{E}_{avg} - D_i \vec{\nabla} n_i + \vec{u} n_i \tag{13}$$

where $\vec{E}_{avg} = \sum_{j=1}^{tf_i} \vec{E}_j / tf_i$. Note the electric field of Eq. (13) is averaged from the duration when solving the simplified continuity equations for ions.

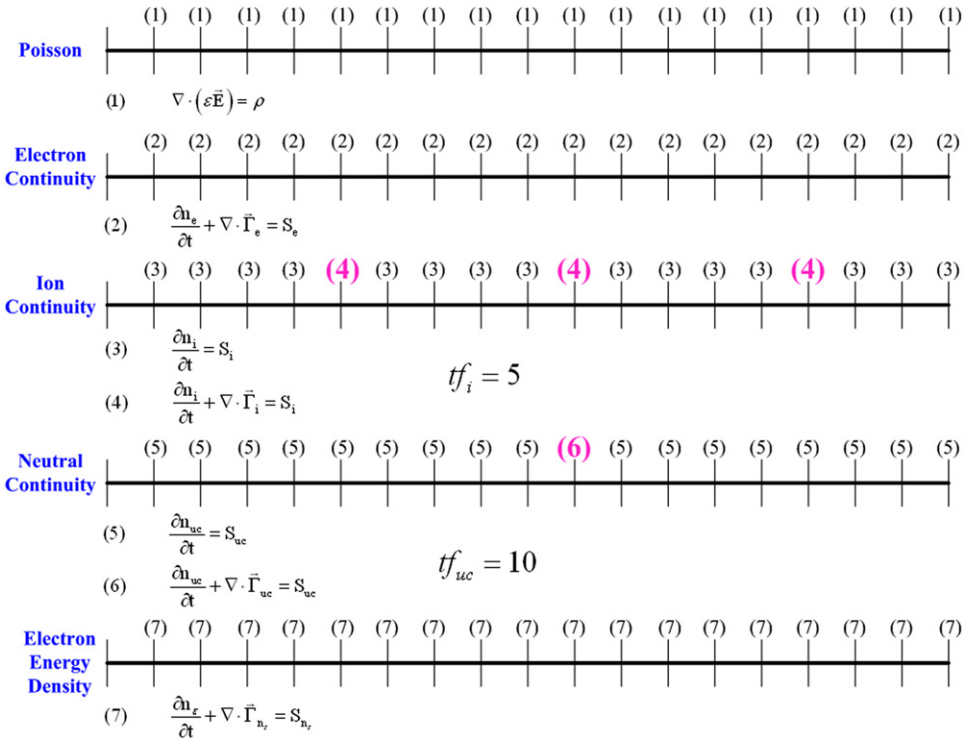

Fig. 3. Time sequence of the temporal multi-scale method.

2.5. Plasma chemistry

In this study, we have employed a realistic dielectric barrier discharge (DBD) jet with helium, consisting of nitrogen impurities to verify and demonstrate the developed numerical approach and its computational efficiency. The helium DBD is simulated in the present study because it is widely used in APPJ applications [1,3–5]. The helium gas flows through the gap between two parallel electrodes and impinges on the substrate with its surroundings filled with air. We consider the helium to contain a trace nitrogen impurity (100 ppm in this study) since it was shown that the discharge current calculated from the pure helium data does not quantitatively agree with experimental results [17]. Although the impurity level is typically less than 0.01%, it was reported that the impurity plays an important role in "pure" helium discharges [18,19]. The simple chemistry set of air (the composition is simplified as 78% N_2 and 22% O_2) is also included for completeness since the surroundings are filled with air for most APPJ applications. In the plasma chemistry, we considered 25 species (e^-, He^+, He_2^+, O_2^+, O^+, O^-, O_2^-, N^+, N_2^+, N_4^+, He_m^*, He_{ex}^*, He_2^*, O_3, O, $O(^1D)$, $O(^1S)$, $O_2(a)$, NO, $N_2(A^3\Sigma_u^+)$, $N_2(B^3\Pi_g)$, $N_2(a'^1\Sigma_u^-)$, $N_2(C^3\Pi_u)$, $N(^2D)$, N) and 101 reaction channels as listed in Table 1. Reaction channels R0–R26 consider chemistry for pure helium discharges. Reaction channels R27–R42 describe the chemistry of oxygen and its interaction with helium, R43–R63 model the chemistry of N_2 and its interaction with helium, and R64–R100 consider the interaction between O_2 and N_2.

The transport coefficients and the rate constants related to the electrons are calculated by solving the Boltzmann equation using BOLSIG$^+$ [27]. Note that these coefficients are predicted and stored in a lookup table as a function of the electron temperature. The mobilities of the ions are taken from the literature [23,28–31], and the corresponding diffusivities are calculated using the Einstein relation. As for the diffusion coefficients of neutral species, those of He_m^*, He_{ex}^*, He_2^*, O_3, O, and N, they are found from the literature [23,29], while that of NO is calculated from the Chapman–Enskog equation for binary diffusion [32], in which the required parameters for calculating the diffusivity can be found from [33]. The diffusivities of excited neutral species are assumed to be equal to those of the corresponding ground-state neutral species since these properties can hardly be found in the literature.

3. Results and discussion

The helium DBD with trace nitrogen (100 ppm) between two parallel electrodes (5 × 50 mm^2) each covered by a ceramic layer (alumina, 1 mm thick) as shown in Fig. 4 is simulated for demonstration of the developed parallel coupling numerical algorithm and the TMSM. The test configuration is similar to that of the experimental study by Chiang et al. [34], except that the length of the electrode becomes 5 mm from the original 50 mm. The helium (30 SLM) flows into the gap (1 mm wide) from the left to the right into an environment filled with air (78% N_2 and 22% O_2) and impinges on the substrate at the right. The resulting Reynolds number based on the gap distance and the average inlet flow speed is 90, which is a typical laminar flow. The background gas species considered in the GFM solver include helium, nitrogen, and oxygen since they determine the local discharge chemistry and their densities are determined by the flow field and the temperature distribution modeled in the GFM solver. The values of relative permittivity for each region employed in this study are $\varepsilon_{r,Discharge}=1.0$, $\varepsilon_{r,Alumina}=11.63$ (measured), $\varepsilon_{r,Teflon}=2.1$ [35] and $\varepsilon_{r,Substrate}=10.0$ for some dielectric material. A realistic nearly sinusoidal voltage with $V_{\text{peak-to-peak}}=6.0$ kV and a frequency of 25 kHz is applied to the electrodes. All simulations were performed on the ALPS supercomputing system at the National Center for High-performance Computing (NCHC) of Taiwan with 2.2 GHz of CPU speed and 2.67 GB of RAM per processor.

Tables 2 and 3 summarize the boundary conditions applied to the PFM and the GFM respectively. The flux of the species continuity equation at the flow inlet (*JA* in Fig. 4) is specified as zero since it is assumed no discharge species flows from the inlet flow, while the flux at the flow outlet (*CD*, and *GH* in Fig. 4) is calculated based on the convection because the species is carried away by

Table 1
Summary of plasma chemistry.

No	Reaction channels	Rate constant or cross- section	Threshold (eV)	Ref.
R00	$e + He \rightarrow e + He$	$BOLSIG^+$	0	[20]
R01	$e + He \rightarrow e + He^*_m$	$BOLSIG^+$	19.82	[20]
R02	$e + He \rightarrow e + He^*_m$	$BOLSIG^+$	20.61	[20]
R03	$e + He \rightarrow e + He^*_{ex}$	$BOLSIG^+$	20.96	[20]
R04	$e + He \rightarrow e + He^*_{ex}$	$BOLSIG^+$	21.21	[20]
R05	$e + He \rightarrow e + He^*_{ex}$	$BOLSIG^+$	22.97	[20]
R06	$e + He \rightarrow e + He^*_{ex}$	$BOLSIG^+$	23.7	[20]
R07	$e + He \rightarrow e + He^*_{ex}$	$BOLSIG^+$	24.02	[20]
R08	$e + He \rightarrow 2e + He^+$	$BOLSIG^+$	24.58	[20]
R09	$e + He^*_m \rightarrow 2e + He^+$	$BOLSIG^+$	4.78	[20]
R10	$e + He^*_m \rightarrow e + He$	$2.9 \times 10^{-15}\ m^3\ s^{-1}$	−19.8	[20]
R11	$e + He^*_2 \rightarrow e + 2He$	$3.8 \times 10^{-15}\ m^3\ s^{-1}$	−17.9	[20]
R12	$He^+ + 2e \rightarrow e + He^*_m$	$6 \times 10^{-32}\ m^6\ s^{-1}$	−4.78	[20]
R13	$He^+_2 + 2e \rightarrow He^*_m + He + e$	$2.8 \times 10^{-32}\ m^6\ s^{-1}$	0	[20]
R14	$He^+_2 + e + He \rightarrow He^*_m + 2He$	$3.5 \times 10^{-39}\ m^6\ s^{-1}$	0	[20]
R15	$He^+_2 + 2e \rightarrow He^*_2 + e$	$1.2 \times 10^{-33}\ m^6\ s^{-1}$	0	[20]
R16	$He^+_2 + e + He \rightarrow He^*_2 + He$	$1.5 \times 10^{-39}\ m^6\ s^{-1}$	0	[20]
R17	$He^*_{ex} + He \rightarrow He^+_2 + e$	$1.5 \times 10^{-17}\ m^3\ s^{-1}$	0	[20]
R18	$He^*_m + He^*_m \rightarrow He^+_2 + e$	$2.03 \times 10^{-15}\ m^3\ s^{-1}$	−18.2	[20]
R10	$He^*_m + He^*_m \rightarrow He^+ + He + e$	$8.7 \times 10^{-16}\ m^3\ s^{-1}$	−15.8	[20]
R20	$He^+ + 2He \rightarrow He^+_2 + He$	$6.5 \times 10^{-44}\ m^6\ s^{-1}$	0	[20]
R21	$He^*_m + 2He \rightarrow He^*_2 + He$	$1.9 \times 10^{-46}\ m^6\ s^{-1}$	0	[20]
R22	$He^*_m + He^*_2 \rightarrow He^+ + 2He + e$	$5 \times 10^{-16}\ m^3\ s^{-1}$	−13.5	[20]
R23	$He^*_m + He^*_2 \rightarrow He^+_2 + He + e$	$2 \times 10^{-15}\ m^3\ s^{-1}$	−15.9	[20]
R24	$He^*_2 + He^*_2 \rightarrow He^+ + 3He + e$	$3 \times 10^{-16}\ m^3\ s^{-1}$	−11.3	[20]
R25	$He^*_2 + He^*_2 \rightarrow He^+_2 + 2He + e$	$1.2 \times 10^{-15}\ m^3\ s^{-1}$	−13.7	[20]
R26	$He^*_2 + He \rightarrow 3He$	$4.9 \times 10^{-22}\ m^3\ s^{-1}$	0	[20]
R27	$e + O_2 \rightarrow e + O_2$	$BOLSIG^+$	0	[21]
R28	$e + O_2 \rightarrow 2e + O^+_2$	$BOLSIG^+$	12.06	[21]
R29	$e + 2O_2 \rightarrow O^-_2 + O_2$	$BOLSIG^+$	0	[21]
R30	$O_3 + O \rightarrow 2O_2$	$8.3 \times 10^{-21}\ m^3\ s^{-1}$	0	[21]
R31	$e + O^+_2 \rightarrow 2O$	$4.8 \times 10^{-13}\ m^3\ s^{-1}$	0	[21]
R32	$O^+ + O_2 \rightarrow O^+_2 + O$	$2.0 \times 10^{-17}\ m^3\ s^{-1}$	0	[21]
R33	$He^*_m + O_2 \rightarrow e + O^+_2 + He$	$2.4 \times 10^{-16}\ m^3\ s^{-1}$	0	[21]
R34	$He^*_m + O \rightarrow e + O^+ + He$	$4.3 \times 10^{-16}\ m^3\ s^{-1}$	0	[21]
R35	$2O + He \rightarrow He + O_2$	$1.04 \times 10^{-45}\ m^6\ s^{-1}$	0	[21]
R36	$O + He + O_2 \rightarrow O_3 + He$	$6.27 \times 10^{-46}\ m^6\ s^{-1}$	0	[21]
R37	$O_3 + He \rightarrow O + He + O_2$	$2.28 \times 10^{-32}\ m^3\ s^{-1}$	0	[21]
R38	$e + O_2 \rightarrow O + O^-$	$BOLSIG^+$	0	[22]
R39	$e + O_2 \rightarrow e + O_2(a)$	$BOLSIG^+$	0.977	[22]
R40	$e + O_2 \rightarrow e + 2O$	$BOLSIG^+$	6.0	[22]
R41	$e + O_2 \rightarrow e + O + O(^1D)$	$BOLSIG^+$	8.4	[22]
R42	$e + O_2 \rightarrow e + O + O(^1S)$	$BOLSIG^+$	10.0	[22]
R43	$He^*_m + N_2 \rightarrow e + N^+_2 + He$	$7.0 \times 10^{-17}\ m^3\ s^{-1}$	0	[23]
R44	$He^*_2 + N_2 \rightarrow e + N^+_2 + 2He$	$7.0 \times 10^{-17}\ m^3\ s^{-1}$	0	[23]
R45	$He^+ + N_2 \rightarrow N^+_2 + He$	$5.0 \times 10^{-16}\ m^3\ s^{-1}$	0	[23]
R46	$He^+ + N_2 \rightarrow N^+ + N + He$	$7.0 \times 10^{-16}\ m^3\ s^{-1}$	0	[23]
R47	$He^+_2 + N_2 \rightarrow N^+_2 + 2He$	$5.0 \times 10^{-16}\ m^3\ s^{-1}$	0	[23]
R48	$He^+_2 + N_2 \rightarrow N^+ + N + 2He$	$7.0 \times 10^{-16}\ m^3\ s^{-1}$	0	[23]
R49	$2e + N^+_2 \rightarrow e + N_2$	$5.651 \times 10^{-39} T_e^{-0.8}\ m^6\ s^{-1}$	0	[23]
R50	$e + N^+_2 \rightarrow 2N$	$2.540 \times 10^{-12} T_e^{-0.5}\ m^3\ s^{-1}$	0	[23]
R51	$e + N_2 \rightarrow e + 2N$	$1.959 \times 10^{-12} T_e^{-0.7} \exp\left(-\frac{1.132\times10^5}{T_e}\right)\ m^3\ s^{-1}$	9.757	[23]
R52	$e + N \rightarrow 2e + N^+$	$8.401 \times 10^{-11} \exp\left(-\frac{1.682\times10^5}{T_e}\right)\ m^3\ s^{-1}$	14.5	[23]
R53	$e + N_2 \rightarrow e + N_2$	$BOLSIG^+$	0.0	[22]
R54	$e + N_2 \rightarrow 2e + N^+_2$	$BOLSIG^+$	15.581	[22]
R55	$e + N_2 \rightarrow e + N_2(A^3\Sigma^+_u)$	$BOLSIG^+$	6.169	[22]
R56	$e + N_2 \rightarrow e + N_2(B^3\Pi_g)$	$BOLSIG^+$	7.353	[22]
R57	$e + N_2 \rightarrow e + N_2(C^3\Pi_u)$	$BOLSIG^+$	11.032	[22]
R58	$e + N_2 \rightarrow e + N_2(a'^1\Sigma^-_u)$	$BOLSIG^+$	8.399	[22]
R59	$e + N^+_4 \rightarrow 2N_2$	$2.0 \times 10^{-12} \left(\frac{T_g}{T_e}\right)^{0.5}\ m^3\ s^{-1}$	0.0	[24]
R60	$N^+_2 + 2N_2 \rightarrow N^+_4 + N_2$	$1.9 \times 10^{-41}\ m^6\ s^{-1}$	0.0	[24]
R61	$N^+_2 + He + N_2 \rightarrow N^+_4 + He$	$1.9 \times 10^{-41}\ m^6\ s^{-1}$	0.0	[24]
R62	$N^+_4 + N_2 \rightarrow N^+_2 + 2\,N_2$	$2.5 \times 10^{-21}\ m^3\ s^{-1}$	0.0	[24]
R63	$N^+_4 + He \rightarrow N^+_2 + He + N_2$	$2.5 \times 10^{-21}\ m^3\ s^{-1}$	0.0	[24]
R64	$N_2(A^3\Sigma^+_u) + N_2(a'^1\Sigma^-_u) \rightarrow e + N^+_4$	$5.0 \times 10^{-17}\ m^3\ s^{-1}$	0.0	[25]
R65	$2\,N_2(a'^1\Sigma^-_u) \rightarrow e + N^+_4$	$2.0 \times 10^{-16}\ m^3\ s^{-1}$	0.0	[25]
R66	$e + N^+_2 \rightarrow N(^2D) + N$	$3.7 \times 10^{-13}\ m^3\ s^{-1}$	0.0	[22]
R67	$e + O^+_2 \rightarrow O + O(^1D)$	$2.1 \times 10^{-13}\ m^3\ s^{-1}$	0.0	[22]

(*continued on next page*)

Table 1 (*continued*)

No	Reaction channels	Rate constant or cross- section	Threshold (eV)	Ref.
R68	$O_2^+ + O_2^- \rightarrow 2O_2$	7.8×10^{-12} m^3 s^{-1}	0.0	[22]
R69	$O^- + N_2^+ \rightarrow O + N_2$	7.8×10^{-12} m^3 s^{-1}	0.0	[22]
R70	$O_2^+ + O^- \rightarrow O + O_2$	7.5×10^{-12} m^3 s^{-1}	0.0	[22]
R71	$O_2^- + O_2(a) \rightarrow e + 2O_2$	2.0×10^{-16} m^3 s^{-1}	0.0	[22]
R72	$O_2^- + N_2(A^3\Sigma_u^+) \rightarrow e + O_2 + N_2$	2.1×10^{-15} m^3 s^{-1}	0.0	[22]
R73	$O_2^- + N_2(B^3\Pi_g) \rightarrow e + O_2 + N_2$	2.5×10^{-15} m^3 s^{-1}	0.0	[25]
R74	$O^- + O_2(a) \rightarrow e + O_3$	3.0×10^{-16} m^3 s^{-1}	0.0	[25]
R75	$O^- + N_2(A^3\Sigma_u^+) \rightarrow e + O + N_2$	2.2×10^{-15} m^3 s^{-1}	0.0	[22]
R76	$O^- + N_2(B^3\Pi_g) \rightarrow e + O + N_2$	1.9×10^{-15} m^3 s^{-1}	0.0	[25]
R77	$O_2^- + O \rightarrow e + O_3$	1.5×10^{-16} m^3 s^{-1}	0.0	[22]
R78	$O^- + O \rightarrow e + O_2$	5.0×10^{-16} m^3 s^{-1}	0.0	[22]
R79	$O^- + N \rightarrow e + NO$	2.6×10^{-16} m^3 s^{-1}	0.0	[22]
R80	$O^- + O_2 \rightarrow e + O_3$	5.0×10^{-21} m^3 s^{-1}	0.0	[25]
R81	$O + N + N_2 \rightarrow NO + N_2$	$1.76 \times 10^{-43} \times T_g^{-0.5}$ m^6 s^{-1}	0.0	[25]
R82	$O + O_2 + N_2 \rightarrow O_3 + N_2$	5.6×10^{-46} m^6 s^{-1}	0.0	[22]
R83	$N_2(A^3\Sigma_u^+) + O_2 \rightarrow 2O + N_2$	1.7×10^{-18} m^3 s^{-1}	0.0	[22]
R84	$N_2(A^3\Sigma_u^+) + O_2 \rightarrow O_2(a) + N_2$	7.5×10^{-19} m^3 s^{-1}	0.0	[22]
R85	$O + N_2(A^3\Sigma_u^+) \rightarrow NO + N(^2D)$	7.0×10^{-19} m^3 s^{-1}	0.0	[22]
R86	$O + N_2(A^3\Sigma_u^+) \rightarrow O(^1S) + N_2$	2.3×10^{-17} m^3 s^{-1}	0.0	[22]
R87	$N_2(B^3\Pi_g) + N_2 \rightarrow N_2(A^3\Sigma_u^+) + N_2$	3.0×10^{-17} m^3 s^{-1}	0.0	[22]
R88	$N_2(B^3\Pi_g) + O_2 \rightarrow 2O + N_2$	1.1×10^{-16} m^3 s^{-1}	0.0	[22]
R89	$N_2(a'^1\Sigma_u^-) + O_2 \rightarrow 2O + N_2$	2.8×10^{-17} m^3 s^{-1}	0.0	[22]
R90	$NO + N_2(a'^1\Sigma_u^-) \rightarrow O + N + N_2$	3.6×10^{-16} m^3 s^{-1}	0.0	[22]
R91	$N_2(C^3\Pi_u) + N_2 \rightarrow N_2(a'^1\Sigma_u^-) + N_2$	1.0×10^{-17} m^3 s^{-1}	0.0	[22]
R92	$N_2(C^3\Pi_u) + O_2 \rightarrow O + O(^1S) + N_2$	3.0×10^{-16} m^3 s^{-1}	0.0	[22]
R93	$N(^2D) + O_2 \rightarrow O + NO$	$1.5 \times 10^{-18} \times (T_g/300)^{0.5}$ m^3 s^{-1}	0.0	[25]
R94	$N(^2D) + O_2 \rightarrow O(^1D) + NO$	6.0×10^{-18} m^3 s^{-1}	0.0	[22]
R95	$O(^1D) + O_2 \rightarrow O + O_2(a)$	3.4×10^{-17} m^3 s^{-1}	0.0	[22]
R96	$O(^1D) + O_2 \rightarrow O + O_2$	$6.4 \times 10^{-18} \times \exp(67/T_g)$ m^3 s^{-1}	0.0	[25]
R97	$N_2(a'^1\Sigma_u^-) \rightarrow N_2 + h\nu(177\text{ nm})$	1.0×10^2 s^{-1}	0.0	[26]
R98	$N_2(A^3\Sigma_u^+) \rightarrow N_2 + h\nu(293\text{ nm})$	0.5 s^{-1}	0.0	[26]
R99	$N_2(B^3\Pi_g) \rightarrow N_2(A^3\Sigma_u^+) + h\nu(1045\text{ nm})$	1.34×10^5 s^{-1}	0.0	[26]
R100	$N_2(C^3\Pi_u) \rightarrow N_2(B^3\Pi_g) + h\nu(336\text{ nm})$	2.45×10^7 s^{-1}	0.0	[26]

T_e is the electron temperature, and T_g is the background temperature. Both are in Kevin.

Table 2
Boundary conditions for the plasma fluid model. The symbols (*A*–*L*) correspond to the vertices in Fig. 4.

	n_e	n_i	n_{uc}	n_ε	ϕ
Solid external domain					
ABC, *DEFG*, *HIJ*	$n_e = 0$	$n_i = 0$	$n_{uc} = 0$	$n_\varepsilon = 0$	$\frac{\partial\phi}{\partial x} = 0$ or $\frac{\partial\phi}{\partial y} = 0$
Flow outlet					
CD, *GH*	$\Gamma_e = 0$	$\Gamma_i = n_i v$	$\Gamma_{uc} = n_{uc} v$	$\Gamma_\varepsilon = 0$	$\frac{\partial\phi}{\partial y} = 0$
Flow inlet					
JA	$\Gamma_e = 0$	$\Gamma_i = 0$	$\Gamma_{uc} = 0$	$\Gamma_\varepsilon = 0$	$\frac{\partial\phi}{\partial x} = 0$
Solid-discharge interface					
ALC, *DG*, *JKH*	[a]	[b]	[c]	[d]	[e]

[a] $\vec{\Gamma}_e = -a \cdot \mu_e n_e \vec{E} - D_e \vec{\nabla} n_e + \frac{1}{4} n_e v_{th}$, where $a = 1$ if drift velocity $(-\mu_e \vec{E})$ points toward the dielectric surface, and $a = 0$ otherwise. The thermal velocity of electron is $v_{th} = \sqrt{\frac{8 k_B T_e}{\pi m_e}}$, where k_B is the Boltzmann constant, and m_e is the electron mass.

[b] $\vec{\Gamma}_i = a \cdot \text{sign}(q_i)\mu_i n_i \vec{E} - D_i \vec{\nabla} n_i$, where $a = 1$ if drift velocity $(\text{sign}(q_i)\mu_i \vec{E})$ points toward the dielectric surface, and $a = 0$ otherwise.

[c] $\vec{\Gamma}_{uc} = -D_{uc} \vec{\nabla} n_{uc}$.

[d] $\vec{\Gamma}_{n_\varepsilon} = 2 k_B T_e \vec{\Gamma}_e$.

[e] $\varepsilon_{dielectric} \vec{E}_{dielectric} \cdot \hat{n} - \varepsilon_{discharge} \vec{E}_{discharge} \cdot \hat{n} = \sigma_f$, where $\hat{n}$ is the unit vector pointing to the wall. σ_f is the surface charge density on the dielectric, which is calculated from the charge species flux on the surface.

Table 3
Boundary conditions for the gas flow model. The symbols (*A*–*L*) correspond to the vertices in Fig. 4.

	u	v	P	T	Y
Solid external domain					
ABC, *DEFG*, *HIJ*	0	0	0	$\frac{\partial T}{\partial X_i} = 0$	0
Flow outlet					
CD, *GH*	$\frac{\partial u}{\partial y} = 0$	$\frac{\partial v}{\partial y} = 0$	$P = 760$ Torr	$\frac{\partial T}{\partial y} = 0$	$\frac{\partial Y}{\partial y} = 0$
Flow inlet					
JA	$u = u_{ini}$ [a]	0	$P = P_{extrapolation}$ [b]	$T = 300$ K	$Y = Y_{ini}$ [a]
Solid–fluid interface					
ALC, *DG*, *JKH*	0	0	$\frac{\partial P}{\partial X_i} = 0$	$k_f \frac{\partial T}{\partial X_i}\Big\|_f = k_s \frac{\partial T}{\partial X_i}\Big\|_s$	$\frac{\partial Y}{\partial X_i} = 0$

[a] The initial velocity u_{ini} and species fraction Y_{ini} are calculated from the specified inlet flow conditions (30 SLM helium gas with 100 ppm nitrogen).

[b] $P_{extrapolation}$ is extrapolated linearly from the pressure of the next two cells.

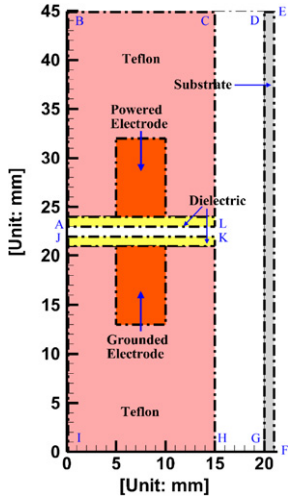

Fig. 4. Sketch of the computational domain.

the background gas flow for heavy species (i.e., ions and neutrals). The boundary conditions of PFM at the solid-discharge interface (*ALC*, *DG*, and *JKH* in Fig. 4) are similar to those boundary conditions applied in [12]. The inlet flow velocity and species fraction are calculated from the inlet flow conditions, and the inlet temperature is assumed as 300 K. The inlet pressure is calculated to balance the pressure difference between flow inlet (*JA* in Fig. 4) and flow outlet (*CD*, and *GH* in Fig. 4) since the pressure at the flow outlet is set as one atmosphere (760 Torr). A Neumann boundary condition is assumed for the flow outlet boundary. Conjugate heat transfer is considered at the solid–fluid interfaces (*ALC*, *DG*, and *JKH* in Fig. 4), and a Neumann boundary condition is assumed for the energy equation at the boundaries of the computational domain.

3.1. Steady-state results of flow field

Fig. 5 shows the converged steady-state results of temperature distribution and flow field obtained from the GFM solver after three iterations with the PFM solver. Iterating further with the PFM does not change the results of the GFM solver significantly, as mentioned previously. As shown in Fig. 5(a), the neutral gas temperature rises from 300 K to 303 K, caused by the gas heating through electron-neutral elastic collision and ionic Joule heating as it passes through the region of parallel electrodes, since this region has the highest discharge intensity. The temperature increase is very small because of the short electrode length. In Fig. 5(b), the maximum flow speed reaches 16 m/s in the gap and the streamlines demonstrate that the flow produces several pairs of vortices above the substrate surface. The complex flow structure affects the distribution of charged and neutral species generated by the DBD, which will be shown later. Further investigation is definitely necessary to reveal the impact of flow structure on the substrate surface for APPJ applications. The converged steady-state results of the GFM solver are adopted for the fluid model simulations in this study.

3.2. Verification of TMSM

Fig. 6 shows the cycle-averaged number densities of electrons, dominant ion (N_4^+), and dominant neutral species ($N_2(A3)$) at the 10th cycle for the cases without (*a*, *c*, *e*) and with (*b*, *d*, *f*) applying the TMSM. The number density of electrons reaches 7×10^{16} m^{-3} in the bulk region of discharge within the parallel electrodes, and the density level in the downstream above the substrate surface drops rapidly down to 10^9–10^{10} m^{-3}. The maximum number density of

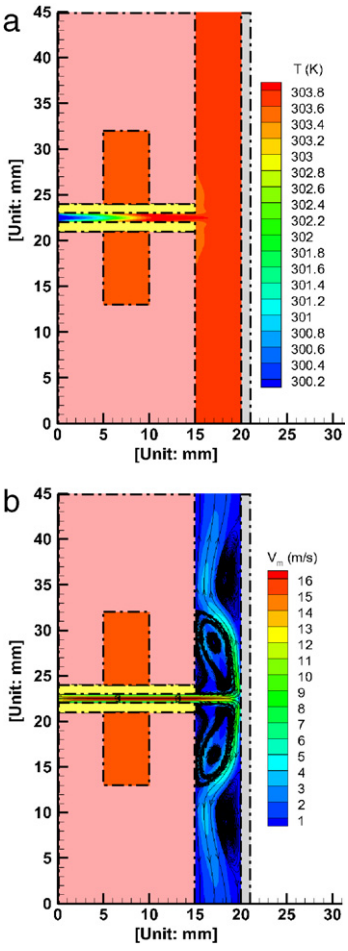

Fig. 5. The converged results of the GFM solver after three interactions with the PFM. (a) Temperature. (b) The $V_m = \sqrt{u^2 + v^2}$ and the streamlines, where u and v are the velocity in the x and y direction respectively.

N_4^+ is slightly higher than that of electrons in the bulk region of discharge within the parallel electrodes, and the density level of N_4^+ in the downstream is also higher than that of electrons. An abundant metastable state $N_2(A3)$ is generated in the discharge region ($\sim 2 \times 10^{18}$ m^{-3}). It is noted that current level of number density represents the "cycle-averaged snapshot" of the transient process and the discharge requires many more time steps to reach the quasi-steady state. As shown in Fig. 6, almost identical distributions of species number densities are obtained for the cases with and without applying the TMSM. Similar results are also obtained for other species, but not shown here. The result clearly verifies the accuracy of the proposed TMSM applied to the PFM.

The simulation of plasma discharge in the PFM needs to solve the Poisson equation, the species continuity equations, and the electron energy density equation. Among these equations, solving the Poisson equation is the most expensive one because it requires the largest number of iterations to obtain the converged solution under the same convergence criteria due to the difficult convergence of elliptic partial differential equation, as discussed in [12]. However, it takes usually most of the runtime for solving the species continuity equations since many species (25 in this case) are often considered for modeling realistic applications. Fig. 7 shows the breakdown of runtime for solving these equations and other computer operations (such as calculating transport properties and rate constants, and communication among processors) for the cases with and without applying the TMSM. The runtime shown in Fig. 7 shows the fractions of runtime for different parts of the code in a typical cycle that is

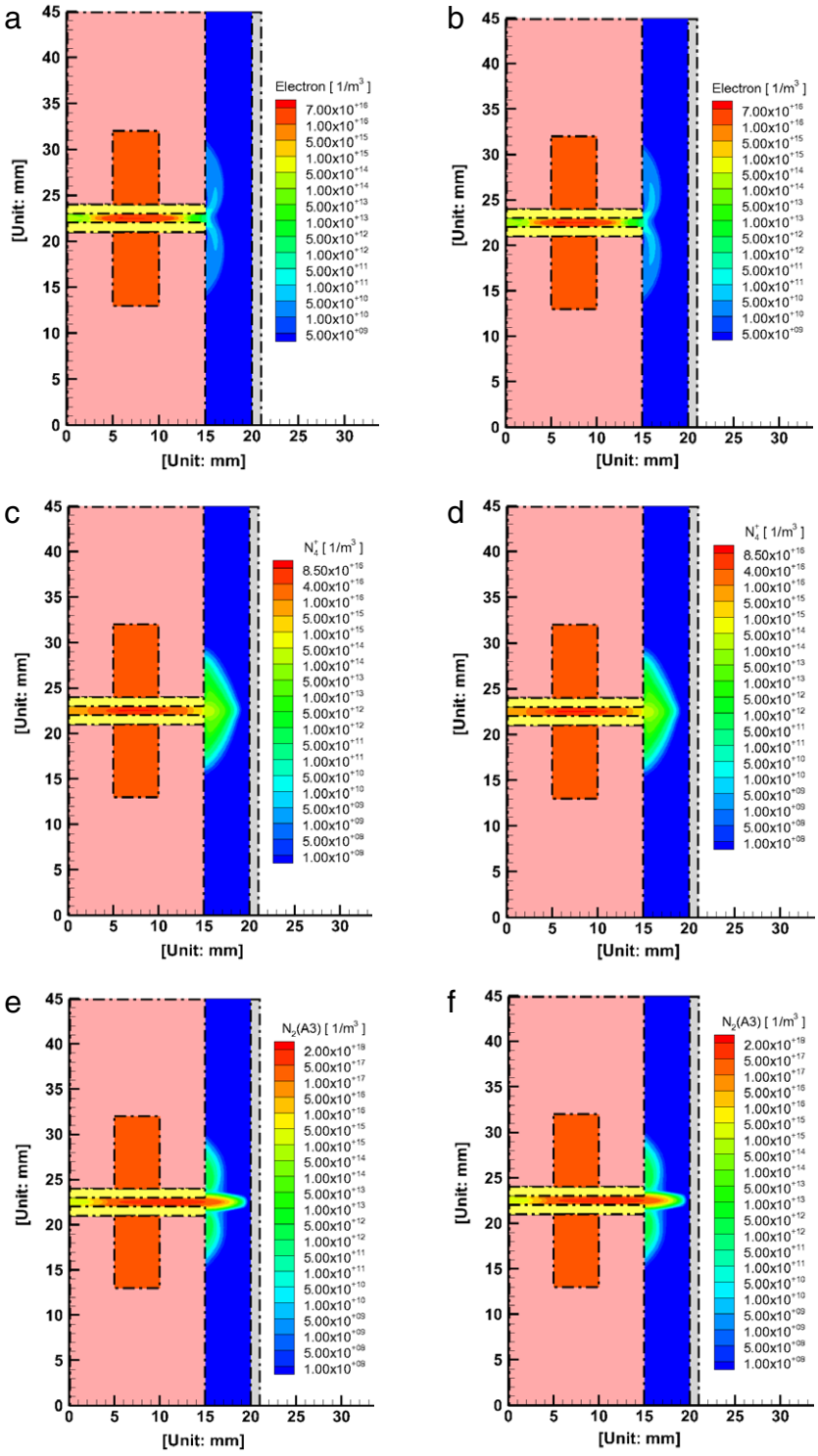

Fig. 6. The cycle-averaged number densities of species at the 10th cycle simulated without (w/o) and with (w) the temporal multi-scale method (a) Electron (w/o), (b) Electron (w), (c) N_4^+ (w/o), (d) N_4^+(w), (e) N_2 (A3) (w/o), (e) N_2(A3) (w).

averaged from five cycles. The control case (without applying the TMSM) shows that it takes more than 6 h for one cycle by using 192 processors and about 55% runtime for solving the species continuity equations. The case with the application of TMSM shows that the runtime for solving the species continuity equations is reduced to only 14%, and it takes about 3.5 h for one cycle by using the same amount of processors (192 processors). The total reduction in runtime is 47% because of the use of the TMSM in this study. Of course, the reduction of runtime increases with an increase of the number of heavy species associated with discharge. Thus, the TMSM is especially useful for simulating discharge with very complex plasma chemistry which involves a large number of heavy species that is not uncommon in practice.

It is obvious that the runtime for solving the continuity equations of heavy species strongly depends upon the size of time factors for ions and neutral species when using the TMSM in PFM. The larger the time factors, the less the runtime is required for solving the continuity equations of heavy species. However,

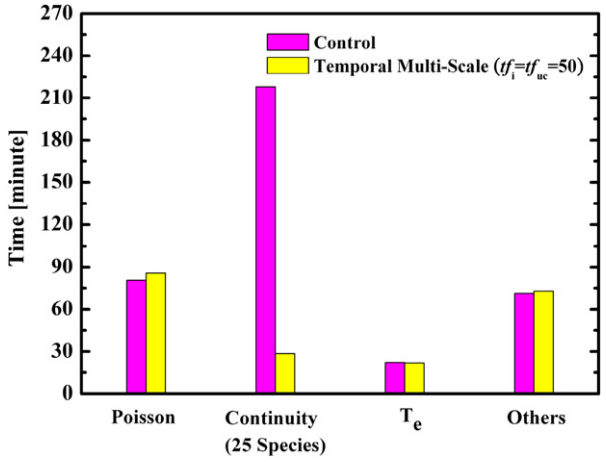

Fig. 7. Breakdown of runtime for solving equations and other computer operations in one cycle. Note that "Poisson", "Continuity", and "Te" represent the Poisson equation, species continuity equations, and the electron energy density equation respectively. The "Others" includes evaluation of transport properties and MPI transmission among processors.

the increase of time factor also raises the numerical error. The allowed maximum time factor of the PFM for applications depends on the transport term of the species continuity equation. In fact, the runtime for solving the species continuity equations reduces close to its minimum as the time factor used is larger than several tens, (e.g. 50). It means that the runtime for solving the species continuity equations with $tf = 50$ is almost equal to that with $tf = 500$ since it takes a much larger portion of the runtime to solve the electron continuity equation at each time step, as compared to the direct time integration of the continuity equations of heavy species. In this study, $tf_i = 50$ and $tf_{uc} = 50$ are used for the case presented in Fig. 7. A further increase of the time factor has little impact on the runtime reduction for solving the continuity equations. Based on some numerical experiments, the values of both tf_i and tf_{uc} can be as large as 1,000 without losing the accuracy though the required runtime is almost equal to that with $tf = 50$, as explained above.

3.3. Demonstration of APPJ simulation

To demonstrate the developed hybrid numerical algorithm is capable of simulating realistic APPJ problems, the cycle-averaged spatial distributions of properties at the 200th cycle are presented in Fig. 8. It is observed that the species number densities reach nearly a quasi-steady state in the region near the substrate. The electron temperature (T_e) of the discharge in the gap between two electrodes is approximately 3–4 eV, which is slightly higher than that of discharges driven by MHz-level atmospheric-pressure discharges [23]. The electron density in the bulk region between electrodes is in the range of 10^{16}–10^{17} m^{-3}, while the density of

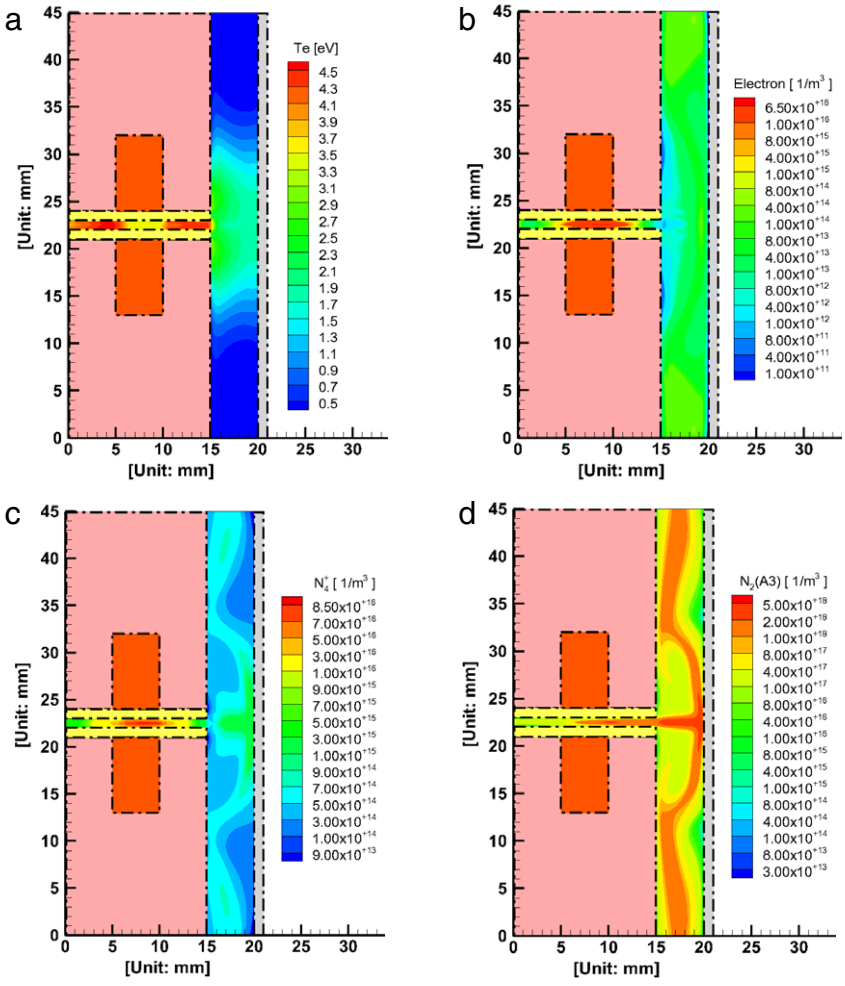

Fig. 8. Cycle-averaged spatial distributions of (a) electron temperature (T_e), (b) number density of electron, (c) number density of N_4^+, (d) number density of N_2(A3) at the 200th cycle.

the dominant ion (N_4^+) has the same level as that of electrons and reaches a maximum of 8.5×10^{16} m^{-3} in the gap. The density of N_4^+ within the gap decreases by one order of magnitude from the edge of electrode (at $x = 10$ mm) to the edge of Teflon (at $x = 15$ mm) due to the decrease of the electric field. After leaving the edge of Teflon, the N_4^+ is carried by the gas flow and impinges on the substrate surface with a density level of 10^{15} m^{-3} in the region near the stagnation point. The density of N_4^+ decreases slowly and maintains its level at 10^{14}–10^{15} m^{-3} along the substrate. The density of N_4^+ within the recirculation zone is relatively lower (1×10^{14}–5×10^{14} m^{-3}) than that right above the substrate. In addition, Fig. 8(d) shows the number density distribution of the metastable N_2(A3), which is also strongly correlated with the background gas flow pattern near the substrate. Note the number density level of long-lived N_2(A3) in the region near the stagnation point can be more than 10^{18} m^{-3}, which is important in various APPJ applications, e.g., Chiang et al. [34]. In other words, the gas flow pattern in the post-discharge region determines the distribution pattern of heavy species, which is the most important region in several APPJ applications. It demonstrates the importance of background gas flow dynamics in APPJ simulations.

4. Conclusion

In the present study, we have proposed a hybrid numerical algorithm which couples weakly the gas flow model (GFM) solver and the plasma fluid model (PFM) solver for simulating the gas flow and discharge of an atmospheric-pressure plasma jet (APPJ). We have employed two numerical approaches to accelerate the computation. The coupling of the GFM solver and the PFM solver is performed by exchanging the results obtained from the steady-state solution (velocities, temperature, and species densities of background gas flow) of the GFM solver with the cycle-averaged solution (the momentum and energy sources) of the PFM solver. Acceleration of the hybrid numerical algorithm include parallel computing of both solvers with domain decomposition using a message passing interface on distributed-memory machines and the temporal multi-scale method (TMSM) for the PFM by taking advantage of the large timescale differences between electron and heavy species. A realistic helium DBD APPJ problem, including trace nitrogen (100 ppm) driven by a power source of $V_{\text{peak-to-peak}} = 6.0$ kV at a frequency of 25 kHz and considering 25 species with 101 reaction channels, is simulated. The coupling starts with the calculation of the PFM, and then the converged solution of the GFM solver is obtained after three iterations between two solvers in the present study.

The TMSM takes advantage of the transport characteristic of heavy species in the PFM. Transport terms in the continuity equations of heavy species are included in the calculations only at a time that is larger than an integer times the electron time step without losing accuracy. Nearly identical results can be obtained for cases with and without applying the TMSM. The total reduction in runtime is 47% if the TMSM is applied to the helium DBD APPJ simulation presented in this study.

The nearly quasi-steady cycle-averaged results obtained at the 200th cycle are presented. The electron and dominant ion (N_4^+) are sustained at the same density level ($\sim 10^{16}$ m^{-3}) in the discharge region between two parallel electrodes, while the dominant neutrals (N_2(A3)) reaches a density level of 10^{18} m^{-3}. It is shown that the distributions of heavy species near the substrate strongly depend upon gas fluid dynamics. The results demonstrate the importance of considering the background gas flow effect in the APPJ simulations. More details of the physical results will be presented elsewhere in the near future.

Acknowledgments

The authors would like to express their sincere thanks to the computing resources provided by National Central University and National Center for High-Performance Computing of Taiwan through NSC-99-2221-E-009-056-MY2 and 97-NSPO(B)-SE-FA04-02.

References

[1] Y. Qiu, Y.J. Hwang, C. Zhang, J. Adhes. Sci. Technol. 16 (2002) 449.
[2] U. Kogelschatz, Plasma Chem. Plasma Process. 23 (2003) 1.
[3] C.H. Yi, Y.H. Lee, G.Y. Yeom, Surf. Coat. Technol. 171 (2003) 237.
[4] Y.F. Hong, J.G. Kang, H.Y. Lee, H.S. Uhm, E. Moon, Y.H. Park, Lett. Appl. Microbiol. 48 (2009) 33.
[5] H.W. Lee, S.H. Nam, A.H. Mohamed, G.C. Kim, J.K. Lee, Plasma Processes Polym. 7 (2010) 274.
[6] Y. Sakiyama, D.B. Graves, Plasma Sources Sci. Technol. 18 (2009) 025022.
[7] J. van Dijk, K. Peerenboom, M. Jimenez, D. Mihailova, J. van der Mullen, J. Phys. D: Appl. Phys. 42 (2009) 194012.
[8] M.J. Kushner, J. Phys. D: Appl. Phys. 38 (2005) 1633.
[9] A.P. Papadakis, G.E. Georghiou, A.C. Metaxas, Plasma Sources Sci. Technol. 14 (2005) 250.
[10] C.T. Hung, M.H. Hu, J.S. Wu, F.N. Hwang, Comput. Phys. Comm. 177 (2007) 138.
[11] C.T. Hung, Y.M. Chiu, F.N. Hwang, J.S. Wu, Comput. Phys. Comm. 182 (2011) 161.
[12] K.M. Lin, C.T. Hung, F.N. Hwang, M.R. Smith, Y.W. Yang, J.S. Wu, Comput. Phys. Comm. 183 (2012) 1225. http://dx.doi.org/10.1016/j.cpc.2012.02.001.
[13] M.H. Hu, J.S. Wu, Y.S. Chen, Comput. Fluids 45 (2011) 241.
[14] J.P. Boeuf, L.C. Pitchford, J. Appl. Phys. 97 (2005) 103307.
[15] Y. Saad, M.H. Schultz, SIAM J. Sci. Stat. Comput. 7 (1986) 856.
[16] X.C. Cai, W.D. Gropp, D.E. Keyes, R.G. Melvin, D.P. Young, SIAM J. Sci. Comput. 19 (1998) 246.
[17] F. Massines, A. Rabehi, P. Decomps, R.B. Gadri, P. Segur, C. Mayoux, J. Appl. Phys. 83 (1998) 2950.
[18] X.H. Yuan, L.L. Raja, Appl. Phys. Lett. 81 (2002) 814.
[19] Y.B. Golubovskii, V.A. Maiorov, J. Behnke, J.F. Behnke, J. Phys. D: Appl. Phys. 36 (2003) 39.
[20] T. Martens, A. Bogaerts, W. Brok, J.V. Dijk, Anal. Bioanal. Chem. 388 (2007) 1583.
[21] D. Lee, J.M. Park, S.H. Hong, Y. Kim, IEEE Trans. Plasma Sci. 33 (2005) 949.
[22] F. Tochikubo, S. Uchida, H. Yasui, K. Sato, Japan. J. Appl. Phys. 48 (2009) 076507.
[23] X.H. Yuan, L.L. Raja, IEEE Trans. Plasma Sci. 31 (2003) 495.
[24] T. Martens, A. Bogaerts, W.J.M. Brok, J.V. Dijk, Appl. Phys. Lett. 92 (2008) 041504.
[25] I.A. Kossyi, A.Y. Kostinsky, A.A. Matveyev, V.P. Silakov, Plasma Sources Sci. Technol. 1 (1992) 207.
[26] M. Capitelli, C.M. Ferreira, B.F. Gordiets, A.I. Osipov, Plasma Kinetics in Atmospheric Gases, Springer, 2000, p. 157.
[27] BOLSIG+. http://www.siglo-kinema.com/bolsig.htm.
[28] L. Mangolini, C. Anderson, J. Heberlein, U. Kortshagen, J. Phys. D: Appl. Phys. 37 (2004) 1021.
[29] J. Waskoenig, K. Niemi, N. Knake, L.M. Graham, S. Reuter, V. Schulz-von der Gathen, T. Gans, Plasma Sources Sci. Technol. 19 (2010) 045018.
[30] H.W. Ellis, R.Y. Pai, E.W. McDaniel, At. Data Nucl. Data Tables 17 (1976) 177.
[31] L.A. Viehland, E.A. Mason, At. Data Nucl. Data Tables 60 (1995) 37.
[32] R.B. Bird, W.E. Stewart, E.N. Lightfoot, Transport Phenomena, John Wiley & Sons, New York, 2007, p. 525.
[33] B.E. Poling, J.M. Prausnitz, J.P. O'Connell, The Properties of Gases and Liquids, McGraw-Hill, New York, 2001, p. B1.
[34] M.H. Chiang, K.C. Liao, I.M. Lin, C.C. Lu, H.Y. Huang, C.L. Kuo, J.S. Wu, C.C. Hsu, S.H. Chen, Plasma Chem. Plasma Process. 30 (2010) 553.
[35] H.F. Mark, D.F. Othmer, C.G. Overberger, Encyclopedia of Chemical Technology, Vol. 11, third ed., Wiley, New York, 1978, p. 14.