

# **PASSKEy**

Parallel Streamer Solver with Kinetics

并行流注等离子体耦合求解器

## **Readme**

使用说明

最后编制日期/Last edited

08/2021

# About PASSKEy

The Parallel Streamer Solver with KinEtics (PASSKEy code) is a Fortran 90 module designed to calculate the spatial-temporal evolution of the electric field, species densities and fluid dynamics in a non-thermal plasma with detailed complex chemistry.

The code is created in Ecole Polytechnique (Laboratoire de Physique des Plasmas, France) by Dr. Yifei ZHU (朱益飞), and improved by Mr. Xiancong CHEN (陈贤聪) with the support of Science and Technology of Plasma Dynamics Laboratory (航空等离子体动力学重点实验室) and the group of Atelier des Plasmas (等离子体计算工坊团队).

Permission to use PASSKEy in non-commercial applications is hereby granted, provided that proper reference is made in publications reporting results obtained using this software. At present, the preferred way to reference PASSKEy is as follows:

**[1] Yifei ZHU, Xiancong CHEN, Yun WU, Svetlana STARIKOVSKAIA: PASSKEy code [software]. Available from <http://www.plasma-tech.net/passkey/> (Science and Technology of Plasma Dynamics Laboratory, Xi'an, China and Laboratoire de Physique des Plasmas, Paris, France, 2021).**

Email: [yifei.zhu.plasma@gmail.com](mailto:yifei.zhu.plasma@gmail.com)

Lark (飞书) group: download the app and join with invitation code **XNREWZPW**

## **Acknowledge of contribution**

Special thanks to Bin ZHANG (Nanjing University of Science Technology), Chenhua REN (Chinese Academy of Sciences) and Longfei LI (Nanjing University Of Aeronautics And Astronautics) in helping with the internal test and improvement of the PASSKEy code.

# Prerequisite

- System / software: **Linux platform + ifort (MKL)**

The solver should be run on a Linux system with Intel Fortran compiler (including MKL). A SRUN or SLURM manage system is recommended.

The free link for ifort and MKL is available on the website of intel (OneAPI HPC kit):

Fortran compiler:

<https://software.intel.com/content/www/us/en/develop/tools/oneapi/hpc-toolkit/download.html?operatingsystem=linux&distributions=webdownload&options=offline>

MKL package:

<https://software.intel.com/content/www/us/en/develop/tools/oneapi/base-toolkit/download.html?operatingsystem=linux&distributions=webdownload>

- Hardware: **HPC single node parallelization**

The use of a high performance computer (HPC) is highly recommended.

The PASSKEy solver is parallelized based mainly on the OpenMP framework, thus the suggested HPC would have as many cores per node as possible.

- An example hardware configuration

CPU: AMD EPYC 7452 (32 cores and 64 processes)

RAM: 256GB DDR4 ECC

## Advantages / Features

- **Accurate.** Directly validated by detailed measurements.
- **Fast.** High computational efficiency due to optimized numerical scheme and parallelization techniques.
- **Functional.** The solver is designed with the potential to solve coupling problems of low temperature plasmas, fluid and combustion (reaction flow).
- **Growing.** The solver is maintained by growing developers in pioneering groups around the world.
- **Support.** The solver is kept supported by national laboratory and leading team.

# Chapter 1 File preparation

- Input files
- Pre-written scripts

- **Input files**

**(1) BOLSIG+ generated file**

This file is generated by the BOLSIG+ package. The file contains the looking up table of reaction rates and swarm parameters defined as functions of E/N (local field approximation) or mean electron energy (local mean energy approximation). The index of the reactions will be later used in the building of PASSKEy.inp file.

**(2) PASSKEy.inp**

This file contains the definition of physical and chemistry conditions. The detailed meaning of each section of this file can be found in Chapter 4, an example can be found in the appendix or the example zip files.

**(3) Coefficients.inp**

This file contains the parameters of the grids configuration and numerical schemes. The detailed meaning of each section of this file can be found in Chapter 5, an example can be found in the appendix or the example zip files.

**(4) Voltage.inp**

This file simply defines the temporal profile of voltage. The time unit is second and voltage unit is volts. Do not give a 0 value at 0 moment. It is not recommended for a fluid model to simulate the inception problem.

- **Scripts**

**(1) Makecase\_run**

Contains the commands to compile and link the code. Remember to set as an executable before running this file.

**(2) Batch.slurm**

The batch file for job submission in the SLURM task management system.

**(3) Batch\_bin.bat**

The batch file run in the windows system. It contains the commands to transform the output ASCII data into Tecplot binary format.

## **Chapter 2 Code preparation and execution**

- Code preparation
- Code compilation
- Code execution

- **Code preparations**

The user has to prepare the following source files for compilation:

(1) The precompiled module files with suffix `.mod` and `.o` in the folder “module”. (19 `.mod` files and 20 `.o` files.)

(2) The F90 source files generated automatically online by the PASSKEy parser. (6 `.f90` files)

(3) The F90 source file templates `udf_XXX.f90`, that has to be defined by user. The user has to specify the discharge geometry and call the calculation command in this way. (2 `.f90` files). An example is given in the package for the users to simulate.

- **Code compilation**

Please copy all the input files, scripts and codes into one folder, and type: `./makecase_run` to generate the executable file `PASSKEy.exe`.

Note: the generated executable cannot be run on the Windows platform (unless you compile the code on windows with the PASSKEy core lib file).

Note: please set the `makecase_run` as executable before running it.

- **Code execution**

Method 1: Type `./PASSKEy`, in this way the code will run immediately.

Method 2: `Sbatch batch.slurm`, in this way the job is submitted to the task manage system.

## **Chapter 3 Post-processing**

- Output subroutines
- Output file formats

The output files are controlled by a set of subroutines. Typical subroutines have been listed in the example `udf_main.f90` source file.

- **Output subroutines**

The output subroutines are stored in the `bin_output` module file, including (but not limited to):

- `output_meshgrid()`
- `output_Potential()`
- `output_density(species_index)`
- `output_Te()`
- `output_charge()`
- `output_Efield_vector()`
- `output_totalenergy_couple()`
- `output_FGHenergy_couple()`
- `output_Efield_ambi_vector()`
- `output_conductivity()`
- `output_source_ionization()`
- `output_voltage()`
- `output_states_fluid()`

- **Output file formats**

The output files exist in various formats, e.g.

- **Mesh.g file:** *the mesh distribution file that can be read directly in Tecplot. To plot this file, just drag this file on the Tecplot icon.*
- **Specie\_XXX.plt:** *the spatial-temporal evolution of species in ASCII format. The user can run a batch command to transform it into a binary. To plot this file, just drag this file on the Tecplot icon.*
- **Efield\_vector.plt:** *the spatial-temporal evolution of electric field in ASCII format. The user can run a batch command to transform it into a binary. To plot this file, just drag this file on the Tecplot icon. The file contains three variables,  $E_x$  (V/m),  $E_y$  (V/m) and  $E/N$  (Td).*
- **Specie\_XXX.q:** *the spatial distribution of species XXX (or other parameters) in binary format at certain time moment. To plot this file, just drag this file on the Tecplot icon together with the mesh .g file.*
- **Efield\_XXX.q:** *the spatial distribution of electric field at XXX time moment. To plot this file, just drag this file on the Tecplot icon together with the mesh .g file.*

- **Material.dat:** the three columns data including the X Y and material index. To plot this file, copy paste the data into Origin or deal with it using GNUplot et al.
- **Voltage.dat:** the two columns data including the Time and Voltage. To plot this file, copy paste the data into Origin or deal with it using GNUplot et al.
- **States\_fluid.plt file:** the spatial-temporal evolution of fluid parameters in ASCII format. The user can run a batch command to transform it into a binary. To plot this file, just drag this file on the Tecplot icon.
- **Couple\_EnergyDep.dat:** the total energy deposition of each species calculated based on  $Q=JE$ . If the species charge is 0 then this value is 0. The file is in X Y value(:) format.
- **Couple\_FGHenergy.dat:** the total fast gas heating energy deposition of each reactions. The first two columns are the X and Y coordinates data, then comes the fractional contribution of each reactions to the total fast gas heating energy, then comes the total fast gas heating energy value and the volume of the collected mesh grid.

The Couple\_EnergyDep.dat and Couple\_FGHenergy.dat file can be directly input into other CFD solvers (e.g. COMSOL) to calculate the fluid dynamics, which is not the focus of PASSKEY.

## **Chapter 4 Parameters details**

- PASSKEy.inp
- Coefficients.inp

- **PASSKEy.inp**

### **(1) materials**

- *Index: the index number of the material. This value should be kept fixed. You can continue add new materials in the list.*
- *Level: the coverage level of a specific material. The cover level should follow the following sequence from upper to lower side: Metal=>dielectric=>plasma=>gas*
- *State: the state of the material. 1 is gas (including plasma) and 2 is solid.*
- *Ismetal: whether or not this material is a metal.*
- *Isdielectric: whether or not this material is a dielectric.*
- *Isreactive: whether or not this material has reactions.*
- *Name: the name of the material.*
- *Epsilon<sub>r</sub>: the relative dielectric permittivity of the material.*
- *Gammasec: the secondary electron emission coefficient of the material.*
- *Gammape: the photo electron emission coefficient of the material (not used yet).*
- *Potential: the initial potential applied on the material (if it is a metal).*
- *Massdensity: the mass density of the material (for calculation of fluid and combustions).*
- *Heatcapacity: the heat capacity of the material (for calculation of fluid and combustions).*

### **(2) circuit**

- *material: the index number of the material that is going to be assigned a circuit leg.*
- *ipowered: whether or not this material is driven by a voltage source.*
- *R: the resistance value connected to the circuit.*
- *L: the inductance value connected to the circuit. (not used yet).*
- *C: the capacitive value connected to the circuit. (not used yet).*

### **(3) plasma**

- *name: the name of the specie*
- *charge: the charge value*
- *Isee: whether this specie leads to secondary electron emission. If so, then we must set corresponding transport parameters in the following section.*
- *iflux: whether to consider the movement of this specie. If consider, then we must set corresponding transport parameters in the following section.*
- *iphoto: whether photoionization contributes to this specie. Please make sure the species with iphoto=1 are in pair thus the sum of charge is always 0.*
- *switch: whether to consider this specie in the calculation.*
- *C, X0, Y0, A0, B0, D: parameters to set the initial distribution of the*

*plasma. The formula writes:*

$$\text{Init\_density} = C \times \exp(-(X-X_0)^2/A_0^2 - (Y-Y_0)^2/B_0^2) + D$$

#### **(4) gas**

- *name: the name of the gas*
- *molegular\_weight: the molecular weight of the gas molecue.*
- *LJ\_energy: the Lenard-Jones energy of the gas molecue.*
- *LJ\_length: the Lenard-Jones length of the gas molecue.*
- *Init\_ratio: the initial molar ratio of the specific gas.*

#### **(5) ION MOBILITY**

- *name: the name of the ion that is going to set mobility values.*
- *value: the mobility of the ion in unit of m<sup>2</sup>/(vs)*

#### **(6) ION DIFFUSION COEF.**

- *name: the name of the ion that is going to set diffusion coefficients.*
- *value: the mobility of the ion in SI unit.*

#### **(7) reactions\_plasma**

- *name: the expression formula of the reaction.*
- *enthapy: the enthalpy of the reaction.*
- *switch: whether to consider this reactions in the system.*
- *iphoto: whether this reaction contributes to the calculation of photoionization strength.*
- *rate\_const: the reaction rate constant. The definition of rate\_const can be a constant, an expression (following fortran format) or in forms of "BOLSIG Cxx". If using the "BOLSIG Cxx" form, the xx value can be found in the BOLSIG+ output file by looking for corresponding reaction index. The unit of the rate\_const is m6/s, m3/s or /s depending on the reaction order.*

- **Coefficient.inp**

**&MPIOMP**

```
num_cores=128,  
num_processes=0
```

/

*Num\_cores* is the number of threads in a single node. This value is used for OPENMP parallelization.

*Num\_processes* is the number of nodes. This value is used for MPI parallelization. In the PASSKEY code, MPI parallelization is only for kinetics calculation in case of huge grids number of huge reaction numbers. Please contact author for this specific requirement.

**&Geometrydata**

```
type_axis=2  
thickness=1.d0  
length_x=5.d-2  
length_y=5.d-2
```

/

*Type\_axis* defines the type of coordinate in the model. 1 means Cartesian coordinate and 2 means a cylindrical coordinate.

*thickness* is the thickness (unit: m) of the model in Z direction. This parameter does not work if *type\_axis* = 2

*length\_x* and *length\_y* are the total length in x and y direction of the computational domain, unit in m.

**&Meshdata**

```
X_refine_start=0.d0  
X_refine_end=0.9d-2  
X_refiner_start=0.d0  
X_refiner_end=5.2d-4  
Y_refine_start=0.d0  
Y_refine_end=1.8d-2  
Y_refiner_start=1.55d-2  
Y_refiner_end=1.71d-2  
X_refine_size=10.d-6  
Y_refine_size=10.d-6  
X_refiner_size=2.5d-6  
Y_refiner_size=2.5d-6  
X_grow_ratio=1.1d0  
Y_grow_ratio=1.1d0  
mesh_mode=1  
ratio_transition=1.01d0  
y_plane_size=2.d-6
```

/

*X\_refine\_start/X\_refine\_end* and *Y\_refine\_start/Y\_refine\_end* define the refinement region (in m) with mesh grid size defined by *X\_refine\_size* and *Y\_refine\_size*.

*X\_refiner\_start/ X\_refiner\_end* and *Y\_refiner\_start/ Y\_refiner\_end* define the refinement region (in m) with mesh grid size defined by *X\_refiner\_size* and *Y\_refiner\_size*.

*X\_grow\_ratio/Y\_grow\_ratio* defines the growing ratio of mesh grid size if the mesh is out of the refining region.

*Mesh\_mode* specifies the optimization mode of meshing. 1 means less grids and faster speed. 2 means the pin electrode can be refined further. It is recommended to give the value of 2 if the user is calculating a pin-to-plane discharge with a sharp pin and large gap.

*Ratio\_transition* defines the transition ratio from the main discharge region approaching the plane electrode.

*y\_plane\_size* defines the mesh grids size near the plane electrode.

The value of *Mesh\_mode*, *Ratio\_transition* and *y\_plane\_size* together help accurately calculating the pin-to-plane discharge.

#### &Kinetic\_options

Kinetic\_implicit=0

Ebarlimit=60.d0

Kinetic\_ILFA=0

Kinetic\_IFGH=1

Kinetic\_asynchronous=0

iBound\_Reaction=1

/

*Kinetic\_implicit* specifies the solver for kinetics calculation. The implicit solver is DVODE and the explicit solver is VODPK. The first one is preferred for very complex chemistry while the second one is recommended as a first trial and for fast calculation.

*Ebarlimit* specifies the maximum mean electron energy value that can appear in the code. In some extreme conditions, the calculated mean electron energy may be too high leading to unphysical results, this option gives a chance to suppress this problem.

*Kinetic\_ILFA* decides the LFA (Local Field Approximation) and LMEA (Local Mean Energy Approximation). 1 means use LFA and 0 means LMEA. Note!!! The BOLSIG+ output file must fit this option.

*Kinetic\_IFGH* decides whether or not to calculate the fast gas heating energy. This option calculates only the fast gas heating energy.

*iBound\_reaction* decides whether to consider kinetics of last cell of plasma region.

Note!!! Please ensure the following combination:

LMEA: Kinetic\_ILFA=0 iBound\_Reaction=1 NewmannAxis\_force=0 cathode\_bc=2

LFA: Kinetic\_ILFA=1 iBound\_Reaction=0 NewmannAxis\_force=1 cathode\_bc=2 or 3

#### &flux\_options

scheme\_swarm=1

scheme\_flux=5

NewmannAxis\_force=0

```
eta_ISG=0.1
cathode_bc=2
iSEE_cathode=0
```

/

*Scheme\_swarm* is a historical parameter. No use currently.

*Scheme\_flux* decides the numerical scheme for electron flux. 1 is Schafetter Gummel method, 2, 3 and 4 are improved SG methods. 5 is the UNO3 method. 4 and 5 are recommended for large scale and high pressure calculation, 1 is recommend for other cases.

*NewmannAxis\_force* enforce the boundary condition of species on the axis. 1 means the number density is set as  $dn/dx=0$ .

*Eta\_ISG* is the coefficient for the ISG method when the *Scheme\_flux* = 2, 3 or 4. This value is usually set as 0.1.

*Cathode\_bc* decides the boundary conditions of the cathode. If LFA is used, this value should be 2 or 3, if LMEA is used, this value should be 2.

*iSEE\_cathode* decides whether to consider the secondary electron emission on the cathode.

&Poisson\_options

```
semimplicit_Poisson=1
Elimit=15000
```

/

*semimplicit\_Poisson* specify whether or not to solve the Poisson equation using semi-implicit method. 1 usually gives faster calculation speed.

*Elimit* decides the maximum electric field in the calculation. The unit is Td.

&Photoionization\_options

```
iphoto=2
kesi=0.1d0
Pquench=60.d0
Po2=152.d0
Xsecmin=0.035d2,
Xsecmax=2.d2
length_absorption=2.d-3
Photo_Ethresh=9000000.d0
Aj_helm=1.986d0,51.d0,4886.d0
nambna_helm=5.53d0,14.6d0,89.d0
```

/

*iphoto* specifies the photoionization model. 1 is the classical integral model (not supported), 2 is the Helmholtz model.

*kesi* specifies the photoionization efficiency.

*Pquench* is the quenching pressure of the radiation specie.

*Po2* is the partial pressure of oxygen if the system is the typical N<sub>2</sub>-O<sub>2</sub> system.

*Xsecmin* is the minimum photoionization cross section for O<sub>2</sub>, unit  $m^{-1}Torr^{-1}$

*Xsecmax* is the maximum photoionization cross section for O<sub>2</sub>, unit  $m^{-1}Torr^{-1}$

*length\_absorption* is the photon absorption length in the mixture, depending on the maximum photoionization cross section, for air it is 0.12 cm. unit m.

*Photo\_Ethresh* the threshold Electric field that photo ionization will be considered. This value is calculated according to ionization and attachment coefficient, and can be obtained by BOLSIG. Unit V/m.

*Aj\_helm* for Helmholtz photoionization model. Fitting parameter for photoionization function. unit in  $\text{m}^{-2}\text{Torr}^{-1}$ .

*nambna\_helm* for Helmholtz photoionization model. Fitting parameter for photoionization function. unit in  $\text{m}^{-1}\text{Torr}^{-1}$ .

&Neutral\_plasma\_options

iforce\_neutral=1

field\_neutral=2

dTlimit\_neutral=5.d-30

Tambipolar=1.d-6

/

*iforce\_neutral* decides whether or not to force neutralization of plasma, by making  $n_e = \sum(n_i)$ . Only works when ambipolar or laplacian field is activated.

*field\_neutral* specify how the electric field is calculated once the discharge enters into neutralization. 1 is linear field (the field changes linearly with the voltage profile) and 2 is the Laplacian field (the field is calculated based on  $j = \sigma E$ ,  $\sigma = \mu n_e$ ).

*dTlimit\_neutral* the timestep below which the plasma is considered entering into neutralization.

*Tambipolar* the time moment to use the ambipolar field for calculation of species flux. Note that the ambipolar field is only for calculation of flux but not for kinetics.

&circuit\_options

icircuit=1

/

*icircuit* is the switch to use circuit module.

&working\_condition

InitP=101325.d0

InitT=300.d0

RiseTime=2.d-9,

DecayTime=2.d-9,

SteadyTime=6.d-9,

PeakVoltage=20.d3

Time\_end=15.d-9

polarity=1

voltageinput=1

/

*InitP* is the initial pressure in unit Pa.

*InitT* is the initial temperature in unit K.

*RiseTime*, *SteadyTime*, *PeakVoltage* and *Time\_end* decides the input voltage profile if *voltageinput*=0.

*Polarity* decides the voltage polarity. 1 is positive and -1 is negative.

*Voltageinput* decides whether to use the voltage profile given in the voltage input file *voltage.dat*.

**&Fluid\_options**

```
ipower=1
ifluid=0
icombustion=0
combustionheat=1
scheme_flux_fluid=1
mode_diffusion=4
iFGH=1
iJoleion=1
Time_start_fluid=0.d0
Time_end_fluid=5.d-3
```

/

*ipower* decides whether to calculate plasma.

*ifluid* decides whether to calculate fluid.

*icombustion* decides whether to calculate reactions when solving fluid equations (not available in this version).

*Time\_start\_fluid* decides the time moment to start the calculation of fluid.

*Time\_end\_fluid* decide the time moment when fluid module is closed.

Note!!! In this version the fluid module is not open. Just keep it as what it is given.

**&Solid\_options**

```
isolid=0
initT_solid=300.d0
```

/

*Isolid* decides whether or not to enable the solid module.

*InitT\_solid* the initial temperature of the solid.

**&Output\_options**

```
output_interval=0.2d-9,
output_logfile=1
output_screen=1
output_mode=2
```

/

*Output\_interval* decides calculation time interval between output files.

*Output\_logfile* no use currently.

*Output\_screen* decides what is shown on the screen. 0 is silent mode, 1 outputs all the information (for HPC track), 2 outputs a progress bar, 3 outputs only the time information. Usually we set this value as 1.

*Output\_mode* decides how the output data is given. 1 means all the results are given in one big file. 2 means the results at different time moment are saved in separated files (for faster download from the HPC).

# **Chapter 5 Geometry definition**

The authors are required to write by hand a small piece of fortran code to define the geometry. The basic idea is to cover the material (defined in the PASSKEY.inp) layer by layer. For example for a pin-to-pin geometry, we have:

```
do i=1,index_max
```

```

x=cell(i)%boundary_cell(4)%X
y=cell(i)%boundary_cell(1)%Y
f(1)=y-1.d4*x**2-1.6d-2
g(1)=y-2*x-1.6d-2
x=cell(i)%boundary_cell(2)%X
y=cell(i)%boundary_cell(1)%Y
f(2)=y-1.d4*x**2-1.6d-2
g(2)=y-2*x-1.6d-2
x=cell(i)%boundary_cell(2)%X
y=cell(i)%boundary_cell(3)%Y
f(3)=y-1.d4*x**2-1.6d-2
g(3)=y-2*x-1.6d-2
x=cell(i)%boundary_cell(4)%X
y=cell(i)%boundary_cell(3)%Y
f(4)=y-1.d4*x**2-1.6d-2
g(4)=y-2*x-1.6d-2

```



These are used to define a parabolic electrode shape in some earlier versions. Please neglect.

```
! 1: set all domain as neutral gas, this material is just used to reduce plasma region to have more speed
```

```
cell(i)%material=>Library_material(5)
```

```
do j=1,4
```

```
cell(i)%boundary_cell(j)%material=>Library_material(5)
```

```
end do
```

```
! 2: cover plasma
```

```
!if
```

```
((cell(i)%boundary_cell(2)%X<=0.2d-2).and.(cell(i)%boundary_cell(1)%Y>=1.9d-2).and.(cell(i)%boundary_cell(3)%Y<=2.35d-2)) then
```

```
cell(i)%material=>Library_material(1)
```

```
do j=1,4
```

```
cell(i)%boundary_cell(j)%material=>Library_material(1)
```

```
end do
```

```
! 3: cover driven electrode over plasma
```

```
if ((cell(i)%Y>=(1.d4/6.d0*cell(i)%X**2+2.25d-2)).and.(cell(i)%boundary_cell(2)%X<=1.d-3)) then
```

```
cell(i)%material=>Library_material(3)
```

```
do k=1,4
```

```
cell(i)%boundary_cell(k)%material=>Library_material(3)
```

```
end do
```

```
end if
```

```
! 4: cover ground electrode over plasma, this is only assigned to the boundary of a inner cell.
```

```
if ((cell(i)%Y<=(-1.d4/6.d0*cell(i)%X**2+2.d-2)).and.(cell(i)%boundary_cell(2)%X<=1.d-3)) then
```

```
cell(i)%material=>Library_material(2)
```

```
do k=1,4
```

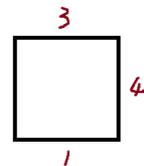
```
cell(i)%boundary_cell(k)%material=>Library_material(2)
```

```
end do
```

```
end if
```

```
end do
```

The index of the mesh boundary edge are given in this way:



## **Chapter 6 Frequently asked questions**

Q1: error #7012: The module file cannot be read. Its format requires a more recent F90 compiler.

A: Please update your ifort version. The code modules are pre-compiled using 19.0.0.117 version. Any ifort newer than this version works. Please find in the previous section the link of the latest fortran compiler and MKL package.

Q2: Can I manually adjust the timestep?

A: The time step is decided automatically according to the CFL time, dielectric relaxation time and kinetics time. It is strongly recommended that you keep the timestep as it is. If you want to manually adjust this value, just add following code (if you want to force the timestep as  $10^{-12}$ s):

```
dtime=1.d-12
```

before following line:

```
call NSSkin_Timestep_synchronous(Time,dTime,i)
```

Q3: Does the code support restarting from a broken moment?

A: Currently not. The data storage is too large.

Q4: at least 5 point needed in one X/Y line.

A: Check mesh settings. In the computational domain of plasma, we must ensure at least 5 nodes in one line thus the interpolation could be conducted.

Q5: How to visualize the data in different output\_mode?

A: If you have `output_mode=2`, then you will have a `.g` mesh file and a set of `.q` data file. What you have to do is just drag these files to the Tecplot icon and adjust the colormap or level set.

If you have `output_mode=1`, you can directly open the file, set time strands in tecplot and play each time moment. If the `.plt` file is too big to open, just run the batch file provided in the example case, then you will have a compressed data file to open.

During calculation, the code may also output some files like: `E_2D.dat`, `material.dat` and `Specie_electron.dat`. These files are debug files recording the materials distribution, field and electron density at the last 50 time step. To visualize these files, drag it to the Tecplot icon to open it, and select `data`→`triangulation` to transform the data into visible version. They are actually X Y value data file, you can also open it directly in the origin and plot contours to see.

## **Chapter 7 SDK (to be opened)**

PASSKEy solver will be opened for distributed development after the first release version is stabilized. The important contribution will be brought into the update release together with the developer's name mentioned.

## **Appendix Error Checklist**

## 1. Errors during online generation

(1) Error: Negative value detected, check coefficients.inp

Solution: Please ensure there is no minus sign “-” after the equal sign “=”.

(2) Error: Refine region larger than the computational region, check coefficients.inp

Solution: Please ensure  $\text{length}_x > X_{\text{refine\_end}}$ ,  $\text{length}_y > Y_{\text{refine\_end}}$ .

(3) Error: Local field approximation/ Local mean energy approximation not in consistency, check.

Solution: Please ensure:

①If  $\text{Kinetic\_ILFA} = 1$ , then  $\text{iBound\_Reaction} = 0$ ,  $\text{NewmannAxis\_force} = 1$ ,  $\text{cathode\_bc} = 2$  or  $3$ , and the heading of the first column of any data segment in “Bolsig+ output.dat” should be  $E/N(Td)$ .

②If  $\text{Kinetic\_ILFA} = 0$ , then  $\text{iBound\_Reaction} = 1$ ,  $\text{NewmannAxis\_force} = 0$ ,  $\text{cathode\_bc} = 2$ , and the heading of the first column of any data segment in “Bolsig+ output.dat” should be  $\text{Energy}(eV)$ .

(4) Error: undefined species in reaction list. Check PASSKEY.inp

Solution: All the species appearing in section **&reactions\_plasma** are defined in **&plasma** or **&materials** section.

(5) Error: undefined ion transport coefficients. Check PASSKEY.inp

Solution: All components with  $\text{iflux} = 1$  in section **&plasma** should be defined in **&ION MOBILITY** and **&ION DIFFUSION**.

(6) Error: Check reaction definitions.

Solution: Please check the definition of reactions in section `&reactions_plasma` in `PASSKEy.inp` and ensure:

① There should be a “^” before “+” to the left of the arrow “=>”.

② There should be a “^” before “++” or “-+”.

(7) No PASSKEy code received in your e-mail.

Solution: Check the trash folder or change another e-mail box (Gmail is temporarily not supported).

## 2. Errors during compilation

**\*If there is an error reported when code compiling, please check the first line message below “./makecase\_run”.**

**(8) Error: -bash: ./makecase\_run: Permission denied.**

Solution: Please set file “makecase\_run” as an executable.

**(9) Error: ./makecase\_run: line 1: ifort: command not found.**

Solution: Intel Fortran compiler is not loaded. Please make sure the ifort compiler is loaded. E.g.

```
source /public1/soft/modules/module.sh
```

```
module load intel/20.4.3
```

\*Please notice that above commands may differ in different platforms. Please consult your informatique team.

**(10) Error: error #7012: The module file cannot be read. Its format requires a mor recent F90 compiler.**

Solution: Please ensure the Intel Fortran compiler is released after 2019.

**(11) Error: ifort: error #10236: File not found: ‘udf\_geometry.f90’**

Solution: Please check whether “auto\_XXX.f90” files, “udf\_XXX.f90” files, “XXX.o” files and “XXX.mod” files are copied into the folder where makecase\_run is located.

**(12) Error: at least 5 points needed in one X line. 3.2d-2, 1.6-2.**

Solution: The number of neighbor grids is less than 5 in the X or Y direction at coordinate (3.2cm,1.6cm).

① Please draw each geometry (gas, plasma, dielectric, metal) defined in udf\_geometry.f90, and check whether there is an acute angle at the border between the plasma region and the dielectric or metal region. If there is, there will be an error. Adjust the plasma region to avoid this condition.

② Check mesh. Please check **&mesh\_data** section in coefficients.input. Check whether the refined mesh region covers the plasma discharge region properly. Sometimes this error can be avoided by refining mesh at the border between plasma region and other region.

**(13) Error:**

```
udf_geometry.f90(50): error #5082: Syntax error, found IDENTIFIER 'THEN' when expecting one of: :: ), : * <END-OF-STATEMENT>; .( / + - ])' ** > PRIVATE
// ...
    if ((cell(i)%X<4.d-3).and.(cell(i)%Y<=5.d-3) then
    -----^
udf_geometry.f90(55): error #6317: An ENDIF occurred without a corresponding IF THEN or ELSE statement.
    end if
    -----^
```

Solution: “udf\_geometry.f90(50): error #5082: Syntax error ...” means the 50<sup>th</sup> line has a syntax error. Please check the syntax errors in udf\_geometry.f90 when this type error happened.

### 3. Errors during code running

(14) Error: forrtl: severe (24): end-of-file during read, unit 10, file /XXXXXX/coefficients.input

Solution: Please check all “XXX.input” files are copied into the folder where PASSKEy.exe locates in.

(15) Error: forrtl: severe (41): insufficient virtual memory

Solution:

If you are using a HPC platform with a scheduling system, please execute the program by submitting the script instead of running on the management node directly.

If you are using a notepad or PC, please decrease the number of mesh because your computer might not have enough memory.

(16) Error:

Failed at t = 5.274755700572741E-009 with idid = 6 at cell 79405 Efield Td is NaN and MU is 1.530155440414508E-002

x = 3.227133423481494E-002 y = 1.790323720965916E-002 density =

Infinity	4091485431706.55	1.453432031860135E+019
3.518486047079642E+018	6.791489860452644E+018	3.789473340852395E+017
3.739031639240567E+017	1.117614090359725E+018	569828873647272.
3.929133807303199E+020	1.015752032042740E+019	9.407013566195454E+018
7.576278644630038E+019	1.058503570844844E+019	6.6666666666667
2.447463768115942E+025	NaN	1.530155440414508E-002
Rate constants =	1.084000000000000E-013	1.153000000000000E-013
8.812000000000000E-017	9.802000000000000E-016	2.353000000000000E-015
6.169000000000000E-016	2.660000000000000E-014	5.000000000000000E-041
5.000000000000000E-041	2.500000000000000E-016	6.000000000000000E-017

## FORTRAN PAUSE

Solution:

**1<sup>st</sup> step:** Understand the error message. The error message indicates there is an error when calculating reaction kinetics at coordinate (3.227cm, 1.790cm) at the time 5.27ns. The Efield (E/N) is NAN (not a number), indicating the electric field calculation in the previous step was also wrong. At this time the electron density (the first value after “density=”) is infinity.

**2<sup>nd</sup> step:** Find the error position. Use e.g. Tecplot to draw Specie\_electron.dat and E\_2D.dat to find the space position where the error occurs. (probably near the electrodes, especially the cathode).

**3<sup>rd</sup> step:** Decide the discharge voltage. Look up for the voltage value at the time of error from voltage.dat, or decide the voltage based on the voltage waveform and the time of error message.

**4<sup>th</sup> step:** Analysis the reason:

(1) If the error occurred during the streamer propagation stage, check the following: a) Is the streamer approaching the boundary of plasma region? If so, adjust the plasma region; b) Is the voltage profile smooth at the error moment? Smooth the voltage if possible; (c) Set the kinetics\_implicit=1 in coefficients.input; (d) Adjust the scheme\_flux (1 or 5 is recommended) in coefficients.input.

(2) If the error occurs when the electrodes are short circuit, and E/N (calculated from simple potential difference between electrodes) far exceeds the gas ionization threshold, the plasma channel will soon turn into an arc, the electron density will increase rapidly and an error

occurs as the LTP model can no longer handle. If in any way you still want to continue the calculation, please follow the steps below:

a) Set `semimplicit_Possion = 1` and recalculate.

b) Recalculate based on LMEA.

\* The calculation results beyond the PASSKEY capacity might not be reliable. The voltage generally drops rapidly when the electrode is short circuit, please use the experimental voltage waveform as the input for the model.

(3) If the error occurs when the electrodes are short circuit, and  $E/N$  **does not** far exceeds the gas ionization threshold, the electron density is moderately high (more than  $10^{21} \text{m}^{-3}$ ) and the time step is relatively long ( $10^{-12} \text{s} \sim 10^{-11} \text{s}$ ), please take following steps:

a) Set `semimplicit_Possion = 0` and recalculate.

b) Set `iSEE_cathode = 1`, adjust cathode boundary condition `cathode_bc` and recalculate.

c) Recalculate based on LMEA.

d) Refine mesh where the error occurs.

(4) If the above methods still failed, try the following options in `coefficients.input`:

a) Set `kinetics_implicit=1`

b) Set `cathode_bc=1`

**If the problems cannot be solved according to this checklist, please send the code file and error message text to:**

[yifei.zhu.plasma@gmail.com](mailto:yifei.zhu.plasma@gmail.com). You can also submit questions in the Gongfang Lark(飞书) group.