



Ion Drift Velocity Benchmark

Introduction

When ions are subjected to a static electric field in a rarefied buffer gas, the combined effect of the electric force and collisions between the ions with ambient gas molecules causes the average ion velocity to approach an equilibrium value known as the ion drift velocity. The ability to accurately predict the drift velocity is important for the operation of devices such as ion mobility spectrometers, which are capable of accurately analyzing gas mixtures containing many different ion species by sorting the different species based on small variations in their drift velocity.

In this example, the average drift velocity of a group of argon ions is computed and compared to experimental data. The ions are modeled using the Charged Particle Tracing interface. Collisions with a neutral background gas are applied using the **Collisions** node, which supports various stochastic collision models including **Elastic** collisions and **Resonant Charge Exchange** reactions. The resulting average ion drift velocity and ion temperature agree closely with values from the literature.

Model Definition

In this example, the model particles are Ar⁺ ions that are released in a neutral background gas of Ar atoms of a specified temperature and number density. The objective is to determine the average steady-state velocity, or *drift velocity*, of the ions under the combined influence of the continuous electric force and discrete collisions with the neutral atoms in the background gas.

The following sections provide an overview of the electric force definition and the theory of Monte Carlo collision modeling with the Charged Particle Tracing interface. For more information on the way random collisions are modeled with the Particle Tracing Module, see the section on Theory for the Charged Particle Tracing interface in the *Particle Tracing Module Users's Guide*.

ELECTRIC FORCE

This model uses a **Parametric Sweep** over the electric field amplitude to analyze its effect on the ion drift velocity. Rather than being specified directly, the electric field amplitude is expressed in terms of the ratio E/N in Townsend units, where E (SI unit: V/m) is the amplitude of the electric field and N (SI unit: $1/\text{m}^3$) is the background gas number density. The Townsend unit is defined as $1 \text{ Td} = 10^{-21} \text{ V m}^2$. In this example, the **Parametric Sweep** is run from a minimum of $E/N = 500 \text{ Td}$ to a maximum of $E/N = 10^5 \text{ Td}$.

The electric force $\mathbf{F}_e$ (SI unit: N) on each particle is simply

$$\mathbf{F}_e = Ze\mathbf{E}$$

where $e = 1.602176565 \times 10^{-19}$ C is the elementary charge and Z (dimensionless) is the charge number. In this example $Z = 1$.

COLLISION TYPES

For the range of energies considered in this benchmark model, the interaction between the simulated Ar⁺ ions and the rarefied background gas of neutral Ar atoms is dominated by the following two reaction types:

- 1 **Elastic** collisions, in which only momentum is exchanged between the model particle and the neutral background argon atom, while the total kinetic energy is conserved; and
- 2 **Resonant Charge Exchange** reactions, in which both charge and momentum are exchanged between the model particle and the background gas, typically resulting in a fast moving neutral atom and an slow-moving ion.

COLLISION FREQUENCY AND COLLISION PROBABILITY

Collisions between the argon ions and the neutral atoms occur at random times based on the total collision frequency $\nu(t)$ (SI unit: 1/s), given by the expression

$$\nu(t) = \sum_j \nu_j(t)$$

where $\nu_j(t)$ is the collision frequency associated with the j th reaction in the model. In this example, $j = 1$ denotes the **Elastic** collision while $j = 2$ denotes the **Resonant Charge Exchange** reaction.

In general the collision frequency associated with a specific collision type is

$$\nu_j = n\sigma_j(\mathbf{v} - \mathbf{v}_g)|\mathbf{v} - \mathbf{v}_g| \quad (1)$$

where

- $\mathbf{v}_g$ (SI unit: m/s) is the velocity of an atom or molecule in the background gas,
- $\mathbf{v}$ (SI unit: m/s) is the particle velocity,
- σ_j (SI unit: m²) is the collision cross section associated with the j th reaction, and
- n (SI unit: 1/m³) is the number density of the gas.

The collision cross section for each reaction type can be an arbitrary function of the relative velocity and is usually determined experimentally. In this example the collision cross

section is approximated as a function of energy using **Analytic** functions, although it can also be interpolated from imported tables of data with the **Interpolation** function type.

The probability $P(t)$ (dimensionless) that a particle undergoes a collision in a time interval $(0, t)$ is

$$P(t) = 1 - \exp\left(-\int_0^t v(t')dt'\right) \quad (2)$$

For constant collision frequency, [Equation 2](#) simplifies to

$$P(t) = 1 - \exp(-vt) \quad (3)$$

COLLISION DETECTION ALGORITHMS

The **Collisions** node in the Charged Particle Tracing interface provides two distinct algorithms for detecting and applying the random collisions between model particles and the atoms or molecules in a rarefied background gas. The choice of algorithm is controlled by the **Collision detection** list, which has options **At time steps taken by solver**, and **Null collision method, cold gas approximation**.

Collision at Time Steps Taken by the Solver

The default option **At time steps taken by solver** is a brute force approach that checks for collisions for each model particle at each time step taken by the solver. The following assumptions are made:

- 1 The collision frequency for each model particle is assumed constant over the following time step taken by the solver, even if the particle is accelerating.
- 2 Each model particle can undergo at most one collision per time step taken by the solver, which is applied at the beginning of the step.

Both assumptions 1 and 2 restrict the size of the time step taken by the solver for the results to be meaningful. The step size must be significantly smaller than the average free time between collisions, which might increase very rapidly if the particles are ions or electrons subjected to a strong electric field.

For each particle, a uniformly distributed random number U is sampled from the interval $(0, 1)$. Then, because the collision frequency is assumed constant over the time step taken by the solver, the logical expression for a collision occurring is obtained by comparing U to the probability from [Equation 3](#),

$$U > \exp(-vt) \quad (4)$$

where some simplification has been made by noting that the probability distribution functions of U and $1 - U$ are exactly the same.

The pseudorandom number U is sampled once for every particle at the beginning of every time step taken by the solver.

Null Collision Method, Cold Gas Approximation

The option **Null collision method, cold gas approximation** is an attempt to improve on the option **At steps taken by solver** by allowing multiple collisions to occur for each particle within a single time step taken by the solver and allowing these collisions to occur at distinct times between the discrete time steps.

The “cold gas approximation” is the omission of the gas velocity term $\mathbf{v}_g$ from Equation 1,

$$\mathbf{v}_j = n\sigma_j(\mathbf{v})|\mathbf{v}|$$

This collision detection algorithm only applies to highly energetic particles, those that move at significantly higher speeds than the thermal speed of the background gas.

This algorithm is called the “null collision method” because the general approach is to overestimate the collision probability over a given time interval, predict collisions based on this overestimate of the collision frequency, and then discard a fraction of collisions based on the extent to which the real frequency is less than this artificial constant value, with these discarded interactions being called “null collisions”. Assume a constant trial frequency v_m , such that $v_m > v(t)$ over the time interval $(0, t)$. If this trial frequency were the actual collision frequency over the time step, then Equation 3 would apply,

$$P_m(t) = 1 - \exp(-v_m t)$$

For each model particle, a trial time t_m (SI unit: s) is computed such that $P_m(t_m) = U_1$, where U_1 is a uniformly distributed number in the interval $(0, 1)$ that is uniquely sampled for each particle. Solving for the trial time yields

$$t_m = -\frac{1}{v_m} \log U_1$$

Now that the trial times for each particle have been computed, a second uniformly distributed random number U_2 , not correlated with U_1 , is also randomly sampled from the interval $(0, 1)$ for each particle that undergoes a collision in the time interval $(0, t)$. If the inequality

$$U_2 \int_0^{t_m} v_m dt > \int_0^{t_m} v(t) dt \quad (5)$$

holds true, then the collision is actually a null collision, and must be discarded.

The first integral in [Equation 5](#) can be computed exactly, because the trial frequency is constant, but that the second integral can only be approximated numerically because the real collision frequency can be an arbitrary function of time. Therefore the size of the time step taken by the solver still has some bearing on the accuracy of the solution, even if this source of error is usually not as large as it is for the option **At time steps taken by solver**.

CHOOSING BETWEEN COLLISION TYPES

For either collision detection algorithm, it is then necessary to determine whether the detected collision is of the elastic or resonant charge exchange variety. The probability p_j of a specific collision type occurring is

$$p_j = \frac{v_j}{v}$$

where the denominator is, as usual, the total collision frequency over all collision types.

REDUCING SIMULATION TIME WITH A STOP CONDITION

The solver sequence in this example uses a **Stop Condition** to terminate the study when the particles have undergone a sufficient number of collisions. The rationale behind specifying a stop condition based on the maximum number of collisions over all particles, is that the maximum time needed for the average ion velocity to reach steady state may vary by several orders of magnitude over the parametric sweep of different electric field amplitudes. Thus, simply running the simulation to the same maximum solution time for every parameter value would be either wasteful for those parameter values that statistically converge most quickly, or inaccurate for those that statistically converge too slowly.

To keep track of the total number of collisions that each particle undergoes, the **Count collisions** check box is selected in the settings for the **Collisions** node. The total number of collisions over all model particles is computed using the expression

`comp1.cpt.cptop1(comp1.cpt.col1.cex1.Nc)`. Here `cpt.cptop1()` is a built-in nonlocal coupling that computes the sum of an expression over all model particles.

Results and Discussion

The relationship between the reduced electric field magnitude and the average ion drift velocity is shown in [Figure 1](#). The average ion temperature is shown in [Figure 2](#).

In both plots, the computed solution shows good agreement with experimental data from [Ref. 1](#).

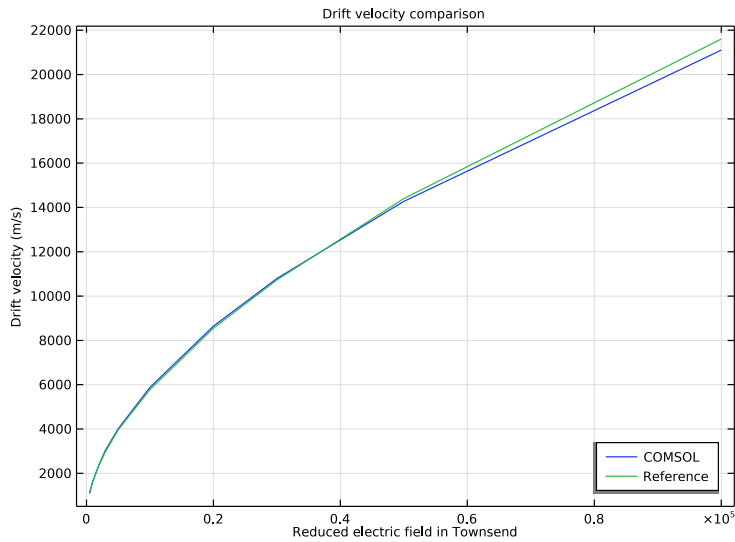


Figure 1: Plot of the drift velocity of Ar^+ ions in a background gas of neutral argon. The average drift velocity is compared to experimental data.

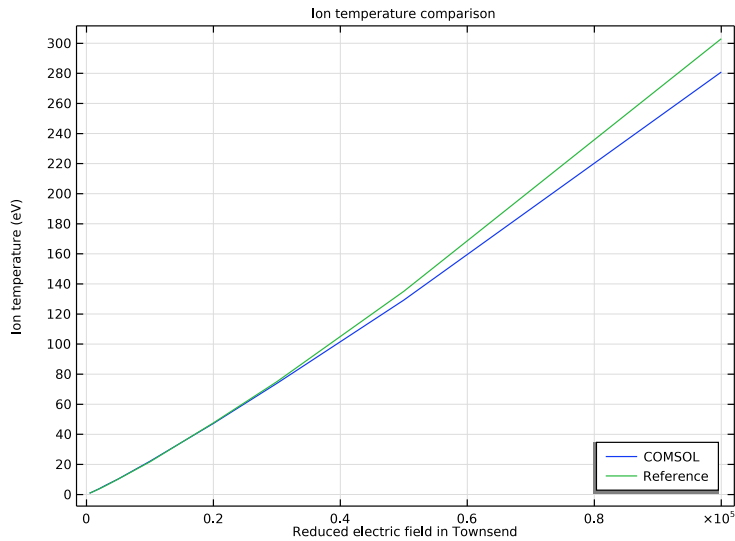


Figure 2: Plot of the ion temperature of Ar^+ ions in a background gas of neutral argon. The average ion temperature is compared to experimental data.

Reference

1. A.V. Phelps, “Cross Sections and Swarm Coefficients for Nitrogen Ions and Neutrals in N₂ and Argon Ions and Neutrals in Ar for Energies from 0.1eV to 10keV,” *J. Phys. Chem. Ref. Data*, vol. 20, no. 3, pp. 557–573, 1991.
 2. A.V. Phelps, “The application of scattering cross sections to ion flux models in discharge sheaths,” *J. Appl. Phys.* vol. 76, pp. 747–753, 1994.
-

Application Library path: Particle_Tracing_Module/
Charged_Particle_Tracing/ion_drift_velocity_benchmark

Modeling Instructions

From the **File** menu, choose **New**.

NEW

In the **New** window, click  **Model Wizard**.

MODEL WIZARD

- 1 In the **Model Wizard** window, click  **3D**.
- 2 In the **Select Physics** tree, select **AC/DC>Particle Tracing>Charged Particle Tracing (cpt)**.
- 3 Click **Add**.
- 4 Click  **Study**.
- 5 In the **Select Study** tree, select **General Studies>Time Dependent**.
- 6 Click  **Done**.

GLOBAL DEFINITIONS

Parameters 1

- 1 In the **Model Builder** window, under **Global Definitions** click **Parameters 1**.
- 2 In the **Settings** window for **Parameters**, locate the **Parameters** section.

3 In the table, enter the following settings:

Name	Expression	Value	Description
mAr	$0.04[\text{kg/mol}] / N_A_const$	6.6422E-26 kg	Ar+ ion mass
ND	$3.2956e22[1/\text{m}^3]$	3.2956E22 1/m ³	Background gas number density
EoverN	500	500	Reduced electric field in Townsend
Ez	$1e-21[\text{V}\cdot\text{m}^2] * EoverN * ND$	16478 V/m	Electric field magnitude
T0	300[K]	300 K	Gas temperature
maxCol	1e5	1E5	Maximum number of collisions for termination

DEFINITIONS

Enter raw data from [Ref. 1](#) for the ion drift velocity as a function of the reduced electric field for elastic collisions between Ar+ ions and neutral Ar atoms.

Interpolation 1 (int1)

- 1 In the **Home** toolbar, click  **Functions** and choose **Local>Interpolation**.
- 2 In the **Settings** window for **Interpolation**, locate the **Definition** section.
- 3 From the **Data source** list, choose **File**.
- 4 Click  **Browse**.
- 5 Browse to the model's Application Libraries folder and double-click the file ion_drift_velocity_benchmark_velocity.txt.
- 6 Click  **Import**.
- 7 Locate the **Units** section. In the **Function** table, enter the following settings:

Function	Unit
Vdrift	m/s

- 8 Locate the **Definition** section. In the **Function name** text field, type Vdrift.

Interpolation 2 (int2)

Enter raw data from [Ref. 1](#) for the ion temperature (eV) as a function of the reduced electric field.

- 1 In the **Home** toolbar, click  **Functions** and choose **Local>Interpolation**.

- 2 In the **Settings** window for **Interpolation**, locate the **Definition** section.
- 3 From the **Data source** list, choose **File**.
- 4 Click  **Browse**.
- 5 Browse to the model's Application Libraries folder and double-click the file `ion_drift_velocity_benchmark_temperature.txt`.
- 6 Click  **Import**.
- 7 Locate the **Units** section. In the **Function** table, enter the following settings:

Function	Unit
ionTemp	eV

- 8 Locate the **Definition** section. In the **Function name** text field, type `ionTemp`.

Enter the analytic approximation for momentum cross section for elastic scattering between Ar⁺ ions and neutral Ar atoms from [Ref. 2](#), which depends on the kinetic energy of the particles.

Analytic 1 (an1)

- 1 In the **Home** toolbar, click  **Functions** and choose **Global>Analytic**.
- 2 In the **Settings** window for **Analytic**, type `Qm` in the **Function name** text field.
- 3 Locate the **Definition** section. In the **Expression** text field, type $1.15e-18 \cdot x^{(-0.1)} \cdot (1+0.015/x)^{0.6}$.
- 4 Locate the **Units** section. In the table, enter the following settings:

Argument	Unit
x	eV

- 5 In the **Function** text field, type `m^2`.

Analytic 2 (an2)

Enter the analytic approximation for isotropic elastic collision between Ar⁺ ions and neutral Ar atoms from [Ref. 2](#), which depends on the kinetic energy of the particles.

- 1 In the **Home** toolbar, click  **Functions** and choose **Global>Analytic**.
- 2 In the **Settings** window for **Analytic**, type `Qi` in the **Function name** text field.
- 3 Locate the **Definition** section. In the **Expression** text field, type $2e-19 / (x^{(0.5)} \cdot (1+x)) + 3e-19 \cdot x / (1+x/3)^{(2.3)}$.

4 Locate the **Units** section. In the table, enter the following settings:

Argument	Unit
x	eV

5 In the **Function** text field, type m^2 .

GEOMETRY I

Cylinder I (cylI)

- 1 In the **Geometry** toolbar, click  **Cylinder**.
- 2 In the **Settings** window for **Cylinder**, locate the **Size and Shape** section.
- 3 In the **Radius** text field, type 2.
- 4 In the **Height** text field, type 3.
- 5 Click  **Build All Objects**.

CHARGED PARTICLE TRACING (CPT)

Particle Properties I

- 1 In the **Model Builder** window, under **Component I (compI)>Charged Particle Tracing (cpt)** click **Particle Properties I**.
- 2 In the **Settings** window for **Particle Properties**, locate the **Particle Mass** section.
- 3 In the m_p text field, type mAr.
- 4 Locate the **Charge Number** section. In the Z text field, type 1.

Wall I

- 1 In the **Model Builder** window, click **Wall I**.
- 2 In the **Settings** window for **Wall**, locate the **Wall Condition** section.
- 3 From the **Wall condition** list, choose **Bounce**.

Release from Grid I

- 1 In the **Physics** toolbar, click  **Global** and choose **Release from Grid**.
- 2 In the **Settings** window for **Release from Grid**, locate the **Initial Coordinates** section.
- 3 In the $q_{z,0}$ text field, type 1.
- 4 Locate the **Initial Velocity** section. From the **Initial velocity** list, choose **Maxwellian**.
- 5 In the N_v text field, type 30.
- 6 In the T_0 text field, type T0.

Electric Force I

- 1 In the **Physics** toolbar, click  **Domains** and choose **Electric Force**.
- 2 In the **Settings** window for **Electric Force**, locate the **Domain Selection** section.
- 3 From the **Selection** list, choose **All domains**.
- 4 Locate the **Electric Force** section. Specify the **E** vector as

0	x
0	y
Ez	z

Collisions I

- 1 In the **Physics** toolbar, click  **Domains** and choose **Collisions**.
- 2 In the **Settings** window for **Collisions**, locate the **Domain Selection** section.
- 3 From the **Selection** list, choose **All domains**.
- 4 Locate the **Fluid Properties** section. In the N_d text field, type ND.
- 5 In the T text field, type T0.
- 6 From the **Collision detection** list, choose **Null collision method, cold gas approximation**.
- 7 Locate the **Collision Statistics** section. Select the **Count all collisions** check box.

Elastic I

- 1 In the **Physics** toolbar, click  **Attributes** and choose **Elastic**.
- 2 In the **Settings** window for **Elastic**, locate the **Collision Frequency** section.
- 3 In the σ text field, type $Q_i(\text{cpt.Ep})$.
- 4 Locate the **Collision Statistics** section. Select the **Count collisions** check box.

Collisions I

In the **Model Builder** window, click **Collisions I**.

Resonant Charge Exchange I

- 1 In the **Physics** toolbar, click  **Attributes** and choose **Resonant Charge Exchange**.
- 2 In the **Settings** window for **Resonant Charge Exchange**, locate the **Collision Frequency** section.
- 3 In the σ text field, type $(Q_m(\text{cpt.Ep}) - Q_i(\text{cpt.Ep})) / 2$.
- 4 Locate the **Collision Statistics** section. Select the **Count collisions** check box.

MESH I

In the **Model Builder** window, under **Component 1 (comp1)** right-click **Mesh 1** and choose **Build All**.

STUDY I

Parametric Sweep

1 In the **Study** toolbar, click  **Parametric Sweep**.

Sweep the reduced electric field from 500 Td to 1e5 Td.

2 In the **Settings** window for **Parametric Sweep**, locate the **Study Settings** section.

3 Click  **Add**.

4 In the table, enter the following settings:

Parameter name	Parameter value list	Parameter unit
EoverN (Reduced electric field in Townsend)	5e2, 1e3, 2e3, 3e3, 5e3, 1e4, 2e4, 3e4, 5e4, 1e5	

Step 1: Time Dependent

To accurately predict the collisions, specify a maximum time and a manual time step size that are proportional to $\sqrt{\text{maxCo1}/\text{EoverN}}$, which in turn is roughly proportional to the free time between collisions.

1 In the **Model Builder** window, click **Step 1: Time Dependent**.

2 In the **Settings** window for **Time Dependent**, locate the **Study Settings** section.

3 In the **Output times** text field, type $0, 5.0e-8 \cdot \sqrt{\text{maxCo1}/\text{EoverN}}$.

Solution 1 (sol1)

1 In the **Study** toolbar, click  **Show Default Solver**.

2 In the **Model Builder** window, expand the **Solution 1 (sol1)** node, then click **Time-Dependent Solver 1**.

3 In the **Settings** window for **Time-Dependent Solver**, click to expand the **Time Stepping** section.

4 From the **Steps taken by solver** list, choose **Manual**.

5 In the **Time step** text field, type $2.5e-9 \cdot \sqrt{\text{maxCo1}/\text{EoverN}}$.

6 In the **Amplification for high frequency** text field, type 1.

- 7 Right-click **Study 1>Solver Configurations>Solution 1 (sol1)>Time-Dependent Solver 1** and choose **Stop Condition**.

Set the stop condition to stop the time-dependent solver when the maximum number of collisions is reached. Make sure to store the solution at the time steps before and after the stop condition is fulfilled.

- 8 In the **Settings** window for **Stop Condition**, locate the **Stop Expressions** section.

- 9 Click  **Add**.

- 10 In the table, enter the following settings:

Stop expression	Stop if	Active	Description
comp1.cpt.sum(comp1.cpt.col1.cex1.Nc)>maxCol	True (>=1)		Stop expression 1

- 11 Locate the **Output at Stop** section. From the **Add solution** list, choose **Steps before and after stop**.

- 12 In the **Study** toolbar, click  **Compute**.

RESULTS

Drift Velocity

- 1 In the **Home** toolbar, click  **Add Plot Group** and choose **ID Plot Group**.

Compare the drift velocity to the tabulated data. Make this comparison at the last time step, after the drift velocity has reached an equilibrium value.

- 2 In the **Settings** window for **ID Plot Group**, type **Drift Velocity** in the **Label** text field.
- 3 Locate the **Data** section. From the **Dataset** list, choose **Study 1/ Parametric Solutions 1 (sol2)**.
- 4 From the **Time selection** list, choose **Last**.
- 5 Locate the **Legend** section. From the **Position** list, choose **Lower right**.

Global 1

- 1 Right-click **Drift Velocity** and choose **Global**.
- 2 In the **Settings** window for **Global**, locate the **y-Axis Data** section.
- 3 In the table, enter the following settings:

Expression	Unit	Description
cpt.ave(cpt.vz)	m/s	
Vdrift(EoverN)	m/s	

- 4 Locate the **x-Axis Data** section. From the **Axis source data** list, choose **Outer solutions**.
- 5 From the **Parameter** list, choose **Expression**.
- 6 In the **Expression** text field, type EoverN.
- 7 Click to expand the **Legends** section. From the **Legends** list, choose **Manual**.
- 8 In the table, enter the following settings:

Legends
COMSOL
Reference

Drift Velocity

- 1 In the **Model Builder** window, click **Drift Velocity**.
- 2 In the **Settings** window for **ID Plot Group**, locate the **Plot Settings** section.
- 3 Select the **y-axis label** check box. In the associated text field, type Drift velocity (m/s).
- 4 Click to expand the **Title** section. From the **Title type** list, choose **Manual**.
- 5 In the **Title** text area, type Drift velocity comparison.
- 6 In the **Drift Velocity** toolbar, click  **Plot**. The resulting image should look like [Figure 1](#).

Ion Temperature

Now compare the ion energy to tabulated data. From [Ref. 1](#), the average ion energy corresponds to a one dimensional energy distribution, that is, $E_p = k_B T / (2e)$.

- 1 Right-click **Drift Velocity** and choose **Duplicate**.
- 2 In the **Settings** window for **ID Plot Group**, type Ion Temperature in the **Label** text field.

Global I

- 1 In the **Model Builder** window, expand the **Ion Temperature** node, then click **Global I**.
- 2 In the **Settings** window for **Global**, locate the **y-Axis Data** section.
- 3 In the table, enter the following settings:

Expression	Unit	Description
cpt.ave(2*cpt.Ep)	eV	
ionTemp(EoverN)	eV	

Ion Temperature

- 1 In the **Model Builder** window, click **Ion Temperature**.

- 2 In the **Settings** window for **ID Plot Group**, locate the **Plot Settings** section.
- 3 In the **y-axis label** text field, type Ion temperature (eV).
- 4 Locate the **Title** section. In the **Title** text area, type Ion temperature comparison.
- 5 In the **Ion Temperature** toolbar, click  **Plot**. The resulting image should look like [Figure 2](#).