AN APPROACH TO ARCJET THRUSTER COMPUTATIONS
FOR FASTER CONVERGENCE

Shigeru Kuchi-ishi* and Michio Nishida1
Department of Aeronautics and Astronautics,
Kyushu University
Fukuoka 812-8581, JAPAN

Abstract

This paper describes an efficient algorithm to reproduce the flow field inside DC arcjet thrusters. The governing equations are the Navier-Stokes equations that have been extended to thermal and/or chemical nonequilibrium flows and Maxwell’s equations together with Ohm’s law. We try to solve them in a finite volume fashion for an infinitely thin three dimensional control volume under the axisymmetric assumption. The numerical flux for the convective term is evaluated using the AUSM-DV (Advection Upwind Splitting Method) scheme. An approximate system of linear equations arising from the linearization with respect to time is approximately solved by the LU-SGS (Lower-Upper Symmetric Gauss-Seidel) algorithm together with the point implicit procedure for flow field equations. GMRES (Generalized Minimal RESidual) algorithm is employed as an electric field solver. The method developed is used to calculate a low-power DC arcjet thruster with nitrogen as propellant and it has been shown that converged result for thermochemical nonequilibrium computation can be obtained within at most several hours by using a standard personal computer. This paper also serves as a detailed description of the numerical method employed in our another study (IEPC-99-029).

Nomenclature

\begin{align*}
A & : \text{Jacobian matrix} \\
A & : \text{area vector} \\
C & : \text{speed of sound} \\
CFL & : \text{CFL number} \\
e & : \text{electron mode} \\
E & : \text{energy per unit volume} \\
f & : \text{nonequilibrium property} \\
F & : \text{vector of inviscid/viscous flux} \\
I & : \text{total current, identity matrix} \\
imax & : \text{maximum index of control volume} \\
n & : \text{time step} \\
neqns & : \text{number of equations} \\
p & : \text{static pressure} \\
Q & : \text{vector of conservative variables} \\
R & : \text{gas constant} \\
R & : \text{residual vector} \\
t & : \text{time} \\
T & : \text{temperature} \\
U & : \text{velocity} \\
v & : \text{velocity} \\
V & : \text{volume, electric voltage} \\
W & : \text{vector of source terms} \\
\rho & : \text{mass density} \\
\sigma & : \text{electrical conductivity} \\
\phi & : \text{electric potential} \\
\tau & : \text{shear stress}
\end{align*}

1. Introduction

Generally, the flow field inside DC arcjet thrusters can be described by the Navier-Stokes equations that have been extended to thermal and/or chemical nonequilibrium gases, plus Maxwell’s equations for electric field. To solve these equations, time-marching algorithms have been widely used since they can apply to both inviscid and viscous flows and can be used in conjunction with virtually any spatial discretization.

Although it has been becoming feasible to solve the equation set numerically owing to advance of the computational environment, the computation tends to be extremely expensive and time consuming. In fact, a number of computational burden will arise if one tries to reproduce the detailed arcjet physics including chemical reactions and a series of transport phenomena, which is indispensable in order to predict the thruster performance accurately.

For a typical flow field in the DC arcjet thruster, the flow is subsonic at the inlet, but is compressible because of density changes induced by volumetric heat addition referred to as Ohmic heating. This is a problem from the computational point of view, since the eigenvalues of the equation system can differ by many orders of magnitude in this low subsonic region, which seriously degrade the convergence property.

The difficulty is also associated with stiff nature of the chemical reaction terms. Since the characteristic time of chemistry is many orders of magnitude smaller than that of the flow field, allowable time increment to integrate the flow field equations...
is forced to be prohibitively small. Hence enormous amount of iterations are required to obtain steady state solutions.

In the present study, an attempt has been made to enhance the convergence property of arcjet thruster computations by overcoming the difficulty described above. Our final goal is to establish reliable numerical methods which enable us to reproduce the basic arcjet physics in sufficiently short operating time.

To what follows, all equations are discretized in a finite volume fashion for an infinitely thin three dimensional control volume under the axisymmetric assumption. Numerical method employed for convective terms is the AUSM-DV (Advection Upwind Splitting Method) scheme. Time integration is performed implicitly by using the LU-SGS (Lower-Upper Symmetric Gauss Seidel) algorithm for the flow field equations, together with the point implicit procedure to relieve stiffness of the thermo-chemical source terms. The GMRES (Generalized Minimal RESidual) algorithm, which solves a linear system of equations efficiently, is employed to solve the electric field equation.

To investigate versatility of the present algorithm, a low-power DC arcjet thruster with nitrogen as propellant is solved as an example.

2. Governing Equations

The equations that describe the flow field inside DC arcjet thrusters are the Navier-Stokes equations that have been augmented to include an equation for each of the gas species and internal energies that are considered. They can be written in conservative form as

$$\frac{d}{dt} \int \int \int Q dV + \int \int F_k dA_k = \int \int W dV$$

(1)

where $Q = [\rho, \rho u, E, f_s]$ is the vector of conservative variables to be solved, and $F$ is the inviscid/viscous flux. The thermo-chemical source vector $W$ may include the Ohmic heating, finite rate mass production, and thermal relaxation terms.

The primary assumptions introduced are 1) the flow is laminar and axisymmetric, 2) the flow field may be in thermal and/or chemical nonequilibrium state, and 3) Lorentz force, hall currents and ion slip effect can be neglected because of very weak magnetic field induced by small arc current. The detailed expressions of the equations set for thermo-chemical nonequilibrium cases are given in [1]. Steady state solutions are obtained as asymptotic solution of the unsteady equations for large time.

The electric field equation can be derived from Ohm’s law and Maxwell’s equations by neglecting the magnetic effect, which results in the scalar elliptic equation with respect to the electric potential $\phi$.

$$\nabla \cdot (\sigma \nabla \phi) = 0$$

(2)

Since characteristic time of the electric field is many orders of magnitude shorter than that of the flow field, it is impractical to solve the electric field equation in a time-marching manner. Hence we set the time derivatives of Eq. (2) to 0 and solve it iteratively at each stage of time integration for the flow field equations.

3. Finite Volume Formulation

The above equation set is discretized by a finite volume approach. Cell-centered scheme is employed, so that all flow properties are defined at center of the control volume. To solve axisymmetric problems, we commonly transfer the governing equations from orthogonal coordinates $(x, y, z)$ to cylindrical coordinates $(r, \theta, z)$ and set the azimuthal derivatives $\partial / \partial \theta$ to zero. This results in an equation set composed of two-dimensional equations and additional source terms which represent the axisymmetric effect.

In the present calculation, instead of using above approach, we discretize the governing equations of a three-dimensional fashion for an infinitely thin three-dimensional control volume under the axisymmetric assumption [2]. The advantage of this discretization is, if one employs a finite volume approach, the scheme is rigorously conservative (thus the free stream can be captured without the error involved in the treatment of metrics terms). This is desirable for internal flow computations, especially for cases where the energy conversion processes are of interest.

Generally, in a finite volume method, the governing equations Eq. (1) are discretized for an arbitrary control volume by the following formula:

$$\frac{V}{\Delta t} \Delta Q + \sum_j (F_k A_k)_j = V \cdot W$$

(3)

where $V$ is the volume of the control volume, and $A$ denotes the area vector of the control surface. The subscript $j$ denotes index of the adjacent control volume. $F_j$ is thus the numerical flux defined at the cell interface. We consider here a control volume $P$, so that the cell 1–2–3–4 on $x$–$y$ plane is rotated by $2\theta$ around $x$ axis as shown in Fig. 1. The physical properties are all defined in center of the control volume. The area vector $(A^z)^\pm$, $(A^n)^\pm$, and $(A^\xi)^\pm$ can be expressed as:

$$(A^z)^+ = (y_3 + y_2) \theta [(y_3 - y_2) \theta, -(x_3 - x_2), 0]$$

$$(A^z)^- = (y_4 + y_1) \theta [(y_4 - y_1) \theta, -(x_4 - x_1), 0]$$

$$(A^n)^+ = (y_3 + y_2) \theta [(y_3 - y_2) \theta, -(x_3 - x_2), 0]$$

$$(A^n)^- = (y_4 + y_1) \theta [(y_4 - y_1) \theta, -(x_4 - x_1), 0]$$

$$(A^\xi)^+ = (y_3 + y_2) \theta [(y_3 - y_2) \theta, -(x_3 - x_2), 0]$$

$$(A^\xi)^- = (y_4 + y_1) \theta [(y_4 - y_1) \theta, -(x_4 - x_1), 0]$$
We have assumed here that $\theta \ll 1$. Since all terms of Eq. (7) is multiplied by $\theta$, it can be set to 1 without losing generality. The expression of the axisymmetric source terms $H$ is much simpler than that for the cylindrical coordinate system, hence the present formulation is also efficient from the computational standpoint.

4. Numerical Flux for Inviscid Terms

We need to evaluate the inviscid flux at the interface of the control volume. Since physical properties are assumed to be piecewisely constant at each control volume, there exists a discontinuity at the interface. This implies that we need to solve the Riemann’s initial-value problem to evaluate the numerical flux. To date, efforts have been made to solve the Riemann problem approximately because exact solution is enormously expensive to obtain.

In the present study, we evaluate the Riemann flux by using the AUSM-DV scheme [3]. AUSM (Advection Upstream Splitting Method) is a hybrid scheme of FDS (Flux Difference Splitting) and FVS (Flux Vector Splitting) in which the advection terms are discretized in a upwind difference manner, whereas the pressure term is split by FVS.

It is theoretically confirmed that AUSM-DV scheme conserves total enthalpy: physical constraint that the total enthalpy is conserved is not broken by the numerical dissipation.

To obtain more than second-order accuracy, we further reconstruct the distribution of the physical properties in the control volume by using MUSCL (Monotone Upstream-centered Scheme for Conservation Laws) interpolation method [4]. While maintaining the same level of accuracy, AUSM + MUSCL approach is much robust and efficient than FDS and less suffered from numerical dissipation than FVS.

5. Time Integration Algorithm

In order to obtain steady state solutions, the spatially discretized governing equations must be integrated in time. We employ here an implicit method, since explicit methods severely deteriorate the convergence property for low-speed flow problems. Using Euler implicit time integration, Eq. (1) can be written in discrete form as

$$\Delta Q^\nu = \frac{\Delta t}{V} R^{\nu+1}$$

(9)

where $R$ is the right-hand side residual and should equal zero for steady state problems. Linearizing Eq. (9) in time yields

$$A\Delta Q^\nu = R^n/V$$

(10)
where
\[ A = I - \frac{\Delta t}{V} \frac{\partial R}{\partial Q} \] (11)

We must invert \( A \) to obtain \( \Delta Q \). Since \( A \) is a large matrix composed of \( \text{neqns} \times \text{inmax} \) elements, direct inversion is enormously expensive. To alleviate this difficulty, \( A \) is inverted approximately. This is possible since \( A \) itself has nothing to do with the steady state solution obtained: the left hand side of Eq. (10) vanishes to zero for steady problems. Hence we can regard \( A \) as an operator which leads the solutions to convergence. We employed here the LU-SGS (Lower-Upper Symmetric Gauss Seidel) method [5] as such an approximate inversion algorithm since it is robust and efficient. In this algorithm, \( A \) is approximately factored as
\[ A \approx (D + L)D^{-1}(D + U) \] (12)
where \( U, L, D \) represent the upper, lower, and diagonal block matrix, respectively. By using this approximation, \( A \) can be inverted without recursive operations.

6. Point–implicit Procedure

The stiffness of the thermo–chemical source terms (\( W \) in Eq. (1)) makes the computation expensive as described before. This comes from the fact that the time scale associated with chemistry tends to be very small compared to the time step required for an effective calculation. Hence \( W \) must be treated in a relaxed way. The point implicit method, firstly proposed by Bussing and Murman [6], eliminates the stiffness problem and allows the solution to be updated with a time step dictated by the CFL condition. In this method, \( A \) is approximately factored as
\[ A = I - \frac{\Delta t}{V} \left( \frac{\partial(R - W)}{\partial Q} + \frac{\partial W}{\partial Q} \right) \approx \left( I - \frac{\Delta t}{V} \frac{\partial (R - W)}{\partial Q} \right) \] (13)

Since the first factor of Eq. (13) is a \( \text{neqns} \times \text{neqns} \) matrix, it can be inverted directly. It is also possible to implement the LU–SGS algorithm without separating \( W \) from the inviscid/viscous Jacobian matrix [7]. This approach, however, may cause numerical instability since the inversion of \( \partial W/\partial Q \) must include the factorization error coming from the approximation of \( A \). If one employs the point implicit procedure, inherent numerical error associated with the matrix inversion is of the order of round–off error at most. This difference is critical for cases where chemical source terms becomes extremely stiff.

7. Electric Field Solver

We need to obtain a converged solution of Eq. (2) per time step of the flow field computation to evaluate the Ohmic heating term in the energy conservation equation of Eq. (1). Since this term plays a critical role to determine flow structure including the arc column, Eq. (2) must be solved with good accuracy. Adding this, it is not the electric potential \( \phi \) itself but the space derivatives of \( \phi \), which is more sensitive to accuracy of the solution, that determines the Ohmic heating. If we evaluate this term by the electric potential distribution with poor accuracy, it tends to fluctuate severely and the flow field equation will never converge. Hence developing an efficient electric field solver for faster convergence is crucial. Equation (2) can be discretized in the same manner as the flow field computation described in section 3, which results in
\[ A\phi = F \] (14)
where \( A \) and \( F \) are the constant matrix and vector, respectively. Again we need to invert the coefficient matrix \( A \) to obtain \( \phi \). One commonly uses a series of relaxation scheme such as Gauss-Seidel, SOR (Successive Line OverRelaxation), or ILU (Incomplete LU factorization) method. These algorithms solve Eq. (14) iteratively. Unfortunately, a significant number of iteration is still required to obtain converged solutions, especially if convergence criterion becomes severe. In this study, \( A \) is inverted by the generalized minimal residual (GMRES) method of Saad and Schultz [8]. This is a generalization of the conjugate gradient method for solving a linear system where the coefficient matrix is not symmetric and/or positive definite. The use of GMRES combined with preconditioning techniques is becoming widespread in many field. We describe the basic procedure of the algorithm below.

To begin with, we set initial guess of the solution vector as \( \phi_0 \), then the first search direction becomes
\[ v_1 = r_0/\|r_0\| \] (15)
where \( r_0 = F - A\phi_0 \) is the initial residual. We now find an approximate solution of Eq. (14) as
\[ \phi = \phi_0 + \sum_{k=1}^{m} a_k v_k \] (16)
In Eq. (16), vectors \( v_k \) constitute a basis for the \( m \)-dimensional Krylov subspace
\[ K_m = \text{span}[r_0, Ar_0, \ldots, A^{m-1}r_0] \] (17)
and coefficients \( a_k \) are chosen so as to minimize the residual norm
\[ J = \|F - A\phi\| = \|r_0 - Az\| \] (18)
where \( z = \sum_{k=1}^{m} a_{k} v_{k} \). Once the first search direction is specified, \( m \)-orthonormal basis vectors \( v_{k} \) can be recursively generated through the Arnoldi process \([8]\). Set \( h_{i,j} \) as
\[
h_{i,j} = (A v_{j}, v_{i})
\]
then the next search direction \( v_{j+1} \) becomes
\[
\hat{v}_{j+1} = A v_{j} - \sum_{i=1}^{j} h_{i,j} v_{j}, \quad j = 1, 2, \ldots
\]
Set
\[
h_{j+1,j} = ||\hat{v}_{j+1}||
\]
then we obtain \( v_{j+1} \) as \( v_{j+1} = \hat{v}_{j+1}/h_{j+1,j} \). Next we want to solve the least squares problem of Eq. (18). It is easily verified that Eq. (18) is equivalent to
\[
J(a) = ||\beta e_{1} - H_{m} a||
\]
Here, \( \beta = ||r_{0}|| \) and the vector \( e_{1} \) is the first column of the \((m+1) \times (m+1)\) identity matrix. \( H_{m} \) is the upper \((m+1) \times m\) Hessenberg matrix whose entries are \( h_{i,j} \). Minimization of Eq. (22) with respect to \( a \) is a well known least squares problem and can be readily solved through QR algorithm.

In case the residual norm Eq. (22) does not drop to a tolerable value, we set \( m := m + 1 \) and seek the additional search direction through the Arnoldi process described above.

GMRES works best when the eigenvalues of matrix \( A \) is clustered. To realize this we apply the preconditioning technique. For reasons described in \([9]\), we choose the right preconditioning of the form
\[
(A P^{-1})(P \phi) = F
\]
The preconditioning matrix \( P \) is an approximation of \( A \) and clusters the eigenvectors of \( A \) around unity. It is worth noting that the only information we need for \( P \) is the products \( a = P^{-1}b \) and this corresponds to that we solve \( A a = b \) approximately. Hence we can consider a number of relaxation schemes as a preconditioner and we applied the SLOR (Successive Line OverRelaxation) algorithm here. Shown in Fig. 2 is the comparison of the present GMRES + SLOR algorithm and conventional SLOR scheme applied to the present electric field equation. The use of GMRES technique dramatically reduce the computational time. It has been shown that the residual is dropped to \( 10^{-9} \) with only 9 search directions.

It is also possible to apply GMRES in Eq. (10) to solve the flow field equations. However, this is less efficient since the time increment \( \Delta t \) is inevitably limited to small due to the strong nonlinearity of the thermo-chemical source term. The application of GMRES becomes advantageous only if the additional computational cost is compensated for by a reduction of the error associated with the approximate factorization for LU-SGS scheme. In the case \( \Delta t \ll 1 \), the approximate factorization works well since the error is basically rounded of the order of \((\Delta t)^{2}\). Hence the advantage of GMRES will be lost.

8. Electron Energy Equation Solver

Assuming thermal nonequilibrium flow, we must solve the electron energy equation to obtain the electron energy distribution. Unfortunately, the equation system becomes extremely stiff when the electron energy is coupled with the flow field equations. This is because of the following:

1. Electron thermal conductivity becomes very large for highly ionized plasma, sometimes scales as \( T_{e}^{5/2} \).
2. The energy exchange rate between the electron and vibrational mode and the energy loss rate due to electron impact dissociation/ ionization \([1]\) tends to be extremely stiff.
3. Since \( T_{e} \) is evaluated from the electron energy defined as \( E_{e} = 3/2 \rho_{e} R_{e} T_{e} \), it tends to fluctuate severely: both \( E_{e} \) and \( \rho_{e} \) are normally very small compared to other flow properties, and they are liable to be suffered from numerical error.

To remedy them, we uncouple the electron energy equation from other flow field equations and solved with respect to the electron temperature. This is in spirit the same idea as \([10]\) where they iteratively solved the electron energy equation in steady form. In the present algorithm, instead of obtaining a converged solution at each time step, we rewrite the equation in delta form. In this case, we do not need to obtain converged solutions of \( T_{e} \) for each time steps, but need to obtain updating value \( \Delta T_{e} \) required for global convergence of the flow field equations. The electron energy equation is discretized in the same fashion as section 3,
which results in

\[ A \Delta T_e = R \]  \hspace{1cm} (24)

where \( A \) is the Jacobian matrix for \( T_e \). For the same reason discussed in section 5, we do not need to invert \( A \) exactly. We solve Eq. (24) using the standard line relaxation method. Furthermore, to prevent excessive variation of \( T_e \), \( \Delta T_e \) is limited by the following:

\[ \Delta T_e = \text{sgn}(\Delta T_e) \cdot \min\{|\Delta T_e|, |\Delta T_e|_{\text{min}}\} \]  \hspace{1cm} (25)

where \( |\Delta T_e|_{\text{min}} = 100 \text{ K} \) is found to work well.

To initiate the calculation, we firstly solve the flow field assuming thermal equilibrium (one temperature calculation), then the converged solution is used as an initial condition of the two temperature calculation.

9. Other Strategies

9.1 Electric voltage updation

If we set the total current as an operating parameter, the electric voltage \( V \) is counted from the electrical potential distribution obtained from the electric field equation. We try to update \( V \) as \( V^{n+1} = V^n + \Delta V^n \) and \( \Delta V^n \) takes the form

\[ \Delta V^n = \omega \left( \frac{I^*}{I^n} - 1 \right) V^n \]  \hspace{1cm} (26)

where \( I \) and \( I^* \) are the total current to be specified and a temporary current evaluated by the new electric potential distribution and \( V^n \). The relaxation parameter \( \omega \) depends on the variation of \( I \), \( I^* \), and \( I^n \) and given as

<table>
<thead>
<tr>
<th>( I^* &lt; I^n )</th>
<th>( I^* &gt; I^n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I^* &lt; I )</td>
<td>( I^* &gt; I )</td>
</tr>
<tr>
<td>( \omega = \omega_0 )</td>
<td>( \omega = 0 )</td>
</tr>
<tr>
<td>( \omega = 0 )</td>
<td>( \omega = 1 )</td>
</tr>
</tbody>
</table>

Table 1: Electric voltage updation

where \( \omega_0 = 0.01 \) is found to be a good choice. This prevents imposing excessive Ohmic heating at the initial stage of the calculation.

9.2 Local time stepping

The time increment \( \Delta t \) in Eq. (26) is determined so that

\[ \Delta t = (CFL) \cdot \min_i \left( \frac{\Delta x}{|V| + C} \right) \]  \hspace{1cm} (27)

The CFL number \( (CFL) \) can be set to more than 1 for an implicit scheme, but is essentially limited to smaller for chemically reacting flows. If we take a view that \( \Delta t \) is just a parameter which controls the convergence property for steady state computations, a simple but effective approach which accelerates the convergence property is the local time stepping. In this approach, time steps which satisfy the local CFL conditions are set variably at each control volumes.

10. Summary of the algorithm

Finally the flowchart of the present algorithm is depicted in Fig. 3.

![Algorithm flowchart](image)

Figure 3: Algorithm flowchart

11. Examples of Computation

Computations were performed for the same nozzle as used in [11] with nitrogen as propellant. The operating conditions were the total current of 6 A and the mass flow rate of 45.4 mg/s. In Fig. 4, computational grid used in the present calculation are shown. The computational domain is divided into 58 cells in the axial direction and 30 cells in the radial direction. Computations are performed using a personal computer with 300 MHz CPU. The translational temperature distribution in the constrictor region for one-temperature calculation is shown in Fig. 5. In Fig. 6, the convergence
The history of the total energy residual and the calculated thrust for one-temperature computation are shown. It was observed that the basic flow structure was obtained at the first 1000 iteration with $CFL = 1$, and then $CFL$ was set to 0.1 to distinguish high-frequency error. Much larger CFL numbers were possible but did not improve the convergence rate. As shown in Fig. 6, the thrust reach a converged value at 5,000 iteration, which corresponds to the CPU time of about 2 hours, and the solution may be regarded as reaching steady state. The most contributing factor for convergence acceleration may be the employment of the local time stepping. The minimum $\Delta t$ was observed just downstream the cathode tip where the temperature and the speed of sound is high. The time step in the vicinity of the anode is an order of magnitude larger than this minimum. Since this region is very viscous in nature, we can raise $\Delta t$ more than allowed by the CFL conditions, hence glob-1 time stepping will deteriorates the convergence rate there.

The residual history for the electron energy equation in two-temperature calculation is shown in Fig. 7. As described in section 8, we firstly solved the flow field as one-temperature calculation to obtain an initial input for the two-temperature calculation. This process is required because we encounter numerical instability if we start the two-temperature calculation from poor initial conditions. In fact many factors associated with the behavior of electron including the ambipolar diffusion and heat conduction sometimes leads to breakdown of the computation. As seen in Fig. 7, the residual of the electron energy rapidly drops to $10^{-3}$ within 1000 iterations, and enters into limit cycle. Although the calculated thrust fluctuates as well, its range is at most of the order of few percent and flow field was successfully reproduced as reported in [1]. Starting from the one-temperature calcu-
lation, we obtained the converged solution of the two-temperature calculation with CPU time of 6 hours.

The computational cost for each procedures per time step is shown in Fig. 8. The evaluation of the thermo-chemical source terms, most of which is the calculation of the mass production rate, is up to half of the total computational time. In contrast, the computational cost for the inviscid flux is found to be only 2.5%. The AUSM-MUSCL approach described in section 4 is much efficient than other approximate Riemann-solvers since it does not include the matrix multiplication arising from the calculation of the characteristic variables. The evaluation of the Jacobian matrix for inviscid terms is most responsible for the computational cost of the time integration process. The matrix-free approach proposed in [12] may reduce the operating time and memory requirement without degrading the convergence rate.

12. Summary

A series of numerical method has been developed which reproduces the thermal and/or chemical nonequilibrium flow field inside DC arcjet thrusters. By applying strategies described, it has become feasible to obtain steady state solutions within the order of several hours with a standard personal computer for a low-power arcjet thruster with nitrogen as propellant, raising the possibility for the numerical simulation to become an efficient and reliable design tool for thruster developments.

References


