THE ELECTRIC PROPULSION INTERACTIONS CODE*

I.G. Mikellides
Science Applications International Corporation
10260 Campus Point Drive, San Diego, CA 92121

M.J. Mandell
R.A. Kuharski
V.A. Davis
B.M. Gardner
Science Applications International Corporation

J. Minor
NASA Marshall Space Flight Center

Abstract

Science Applications International Corporation is currently developing the Electric Propulsion Interactions Code, *EPIC*, as part of a project sponsored by the Space Environments and Effects Program at the NASA Marshall Space Flight Center. Now in its second year of development, *EPIC* is an interactive computer tool that allows the construction of a 3-D spacecraft model, and the assessment of a variety of interactions between its subsystems and the plume from an electric thruster. These interactions may include erosion of surfaces due to sputtering and re-deposition of sputtered materials, surface heating, torque on the spacecraft, and changes in surface properties due to erosion and deposition. This paper describes the overall capability of EPIC and provides an outline of the physics and algorithms that comprise many of its computational modules.

Introduction

More than one hundred satellites are currently operating in space with onboard electric propulsion (EP). Many more missions with EP systems - both near-Earth and deep space - are being considered. A critical engineering issue in the integration of EP on spacecraft is the potentially unfavorable interaction of their high-energy plumes with surrounding critical components and diagnostic equipment. Such interactions may affect mission lifetime and sometimes even threaten mission success. As NASA considers more powerful EP technologies for long-duration missions, the need to predict pertinent interactions becomes even more critical. Only a few programs with the goal to produce large-scale, 3-D, global computer tools for the assessment of such interactions exist - two in the U.S.[1,2] and one in Europe.[3] One of the two U.S. programs aims in the development *EPIC*.

EPIC is currently developed by Science Applications International Corporation (SAIC) as part of a 2-year effort that is sponsored by the Space Environment and Effects (SEE) Program at

* Major portions of this paper have been presented at the AIAA 38th and 39th Joint Propulsion Conferences (AIAA paper numbers 2002-3667 and 2003-4871)

the NASA Marshall Space Flight Center (MSFC). The objective is to produce an integrated modeling package that can be used to aid the design of spacecraft with onboard electric propulsion. The suite of computer tools is intended to allow both expert and novice users to investigate EP plume-S/C interactions. The code is built on the Module Integrator and Rule-based Intelligent Analytic Database (MIRIAD) architecture, which has formed the core of several NASA and Department of Defense programs.

Overview

In general, the spacecraft designer provides the following input to *EPIC*: satellite geometry and surface materials, thruster locations and plume parameters, case study parameters such as sputter yield coefficients, orbit and hours of thruster operation. The output may be in the form of contour plots of the plume map in space and of surface interactions on the 3-D spacecraft, 1-D plots along surfaces (e.g. erosion depth on a solar array as a function of distance from the thruster), integrated results over duration of mission (e.g. total induced torque in a given direction, total deposition of eroded material at a specific location on the spacecraft), list of results in text format for post-processing. *EPIC*'s various components have already been used to asses EP-spacecraft interactions for both the government[1,4] and industry.[5]

The computer tool (built for the Win32 platform) consists of the following main components:

- An enhanced version of the 3-D geometry-definition tool Object Toolkit (OTk) to facilitate interactive generation of the spacecraft geometry and materials.
- A 3-D interactions code built using the Module Integrator and Rule-based Intelligent Analytic Database (MIRIAD) architecture.
- A 2-D plume code that generates the plume map from an electrostatic thruster in the R-Z plane. A fast, 1-D Hall effect thruster (HET) algorithm is also part of the plume modeling capability. The algorithm is intended to provide a more quantitative estimate of the conditions near the thruster exit.

EPIC Architecture

The bulk of *EPIC*'s capability is the 3-D plume-surface interactions code. The code is built on the MIRIAD architecture, which has formed the core of several NASA and DoD programs, and has allowed spacecraft designers to quickly and affordably model interactions between the spacecraft and the space environment. MIRIAD's open architecture provides a framework for integrating a variety of physical models and their constituent data into a single executable application. The application then allows the user to define systems of interest and perform parametric studies with relative ease. Ideally, any variable in MIRIAD can be included in a parametric study.

The main *EPIC* user interface and the code that integrates the various components are programmed in C# (C-sharp). C# is a new language developed by Microsoft that provides an easy-to-use object orientated way to program the windows user interface, and makes it extremely easy to combine components written in different languages using either the new Common

Language Runtime (CLR) or the Component Object Model (COM) that has been the backbone of windows programs for years. The combination of C# and the CLR foundation classes is a very powerful and easy way to get the desired functionality in the user interface.

The most natural and powerful way to communicate with the MIRIAD data server that coordinates *EPIC* calculations is through COM. It was therefore particularly important to be able to use COM components in *EPIC*. Any COM or dot net client can access *EPIC*'s COM data server. For example EXCEL can be used to script calculations and manipulate data produced by *EPIC*. C# allows easy use of our COM- based 3-D viewer (MxOpenGlArena) for viewing the system and displaying surface information and spatial data.

Main Computation Components of *EPIC*

Spacecraft model

Object Toolkit is *EPIC*'s 3-D object generator. Originally developed to create spacecraft surface models for the *Nascap-2k* plasma interactions code,[6] OTk has been generalized to allow for the definition of spacecraft models for other analysis codes. In particular for *EPIC*, employment of "Special Objects" has been allowed to facilitate the specification of thrusters as illustrated in Figure 1. "Special Objects" are objects understood by the target application that are included in Object Toolkit primarily to specify and display their location. They are therefore not a part of the spacecraft surface model. Each special object instance has an object type name (*e.g.* "Hall Thruster"), an instance name (*e.g.* "SPT100-1"), and a set of properties that includes, at minimum, its position ("x", "y", "z"), thrust unit vector, and the color with which it is to be displayed ("r", "g", "b"). The properties can be edited for each special object instance in similar fashion to the attribute property values. Figure 2 shows the OTk spacecraft model of Figure 1 imported into the main *EPIC* interface, and the list of all materials associated with the spacecraft.

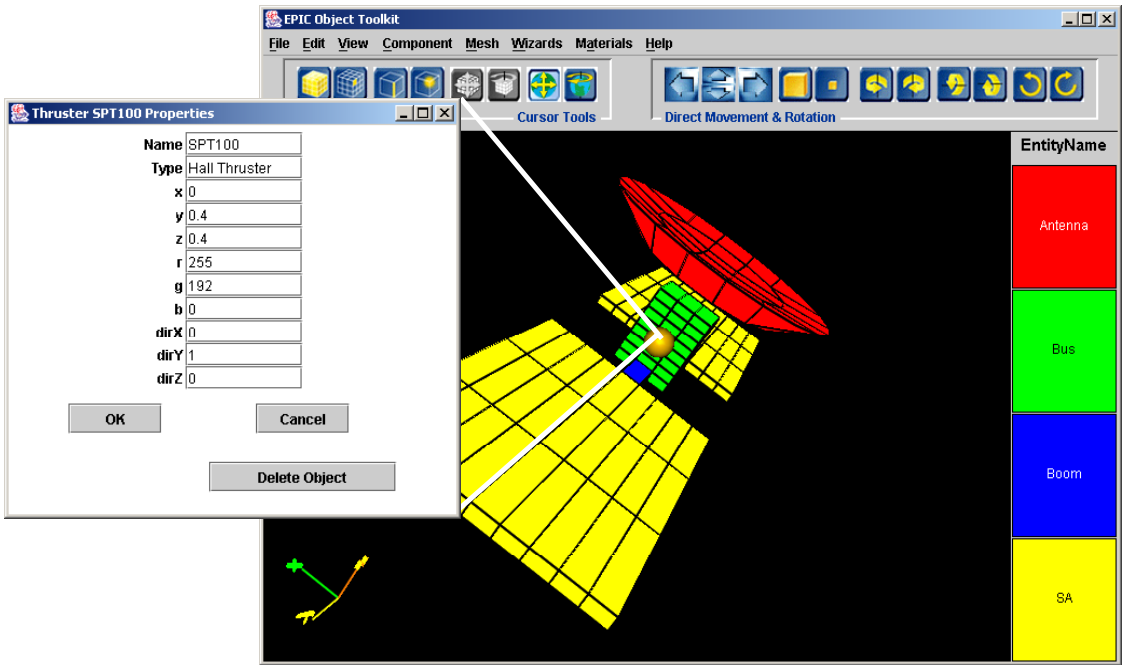

Figure 1. Object ToolKit-generated spacecraft showing various entity definitions and the thruster (in this case a Stationary Plasma Thruster, SPT-100) location.

Electric propulsion plumes

Thruster plume definition in *EPIC* is performed using "PlumeTool" (Figure 3). The 2-D plume generation computational module allows the spacecraft designer to either import a plume map (produced by other codes) or generate it using the existing 2-D plume code. The 2-D plume code was written to model low-density EP exhausts such as those associated with electrostatic thrusters. The code uses parts from an existing, finite element (FE) code (*Gilbert*) that has been used in the past to solve a variety of problems involving Poisson's equation (e.g. see Mikellides, I.G., *et al.*[7]). The generated plume maps consist of the spatial distributions of ion density and velocity. The model consists of three main components: a Lagrangian algorithm for determining the expansion of the main ion beam, a Particle-in-Cell (PIC) solver for computing the dynamics of the charge-exchange plasma, and an ideal-scattering algorithm that calculates the density and velocity of ions that are produced by elastic collisions with neutral atoms. The physics and numerics of each component have been described in detail in previous publications.[1,8,9,10] Highlighted below are some of the fundamental principles behind the various plume model components.

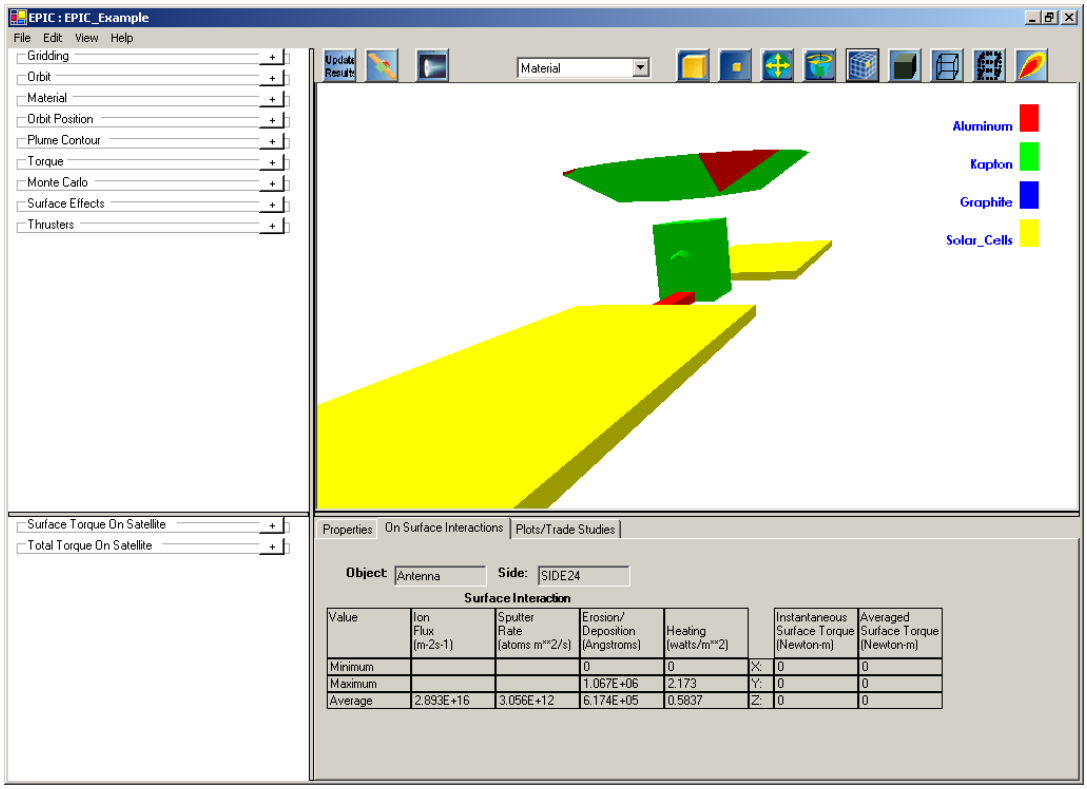

Figure 2. Main *EPIC* interface showing imported spacecraft object and thruster properties.

2-D Plume Physics Models. The main ion beam is assumed to be a collisionless, singly-ionized, quasi-neutral plasma expanding under the influence of the electric field. The latter is assumed to exist due to the presence of density gradients in the plasma. By comparison to heavy-particle motion, electrons reach dynamic equilibrium at much smaller characteristic times. The electron inertia term may therefore be neglected in the equation of motion. In the absence of electron-ion collisions and magnetic fields, integration of the conservation of momentum for the electrons leads to the Boltzmann relation, which can be expressed in terms of the electric potential as follows:

$$\phi(n, T_e) = T_e \ln(n / n_\infty) \qquad (1)$$

where, ϕ is the electric potential, n is the plasma density ($n=n_e=n_i$), T_e is the electron temperature and n_∞ is the reference plasma density at zero potential.

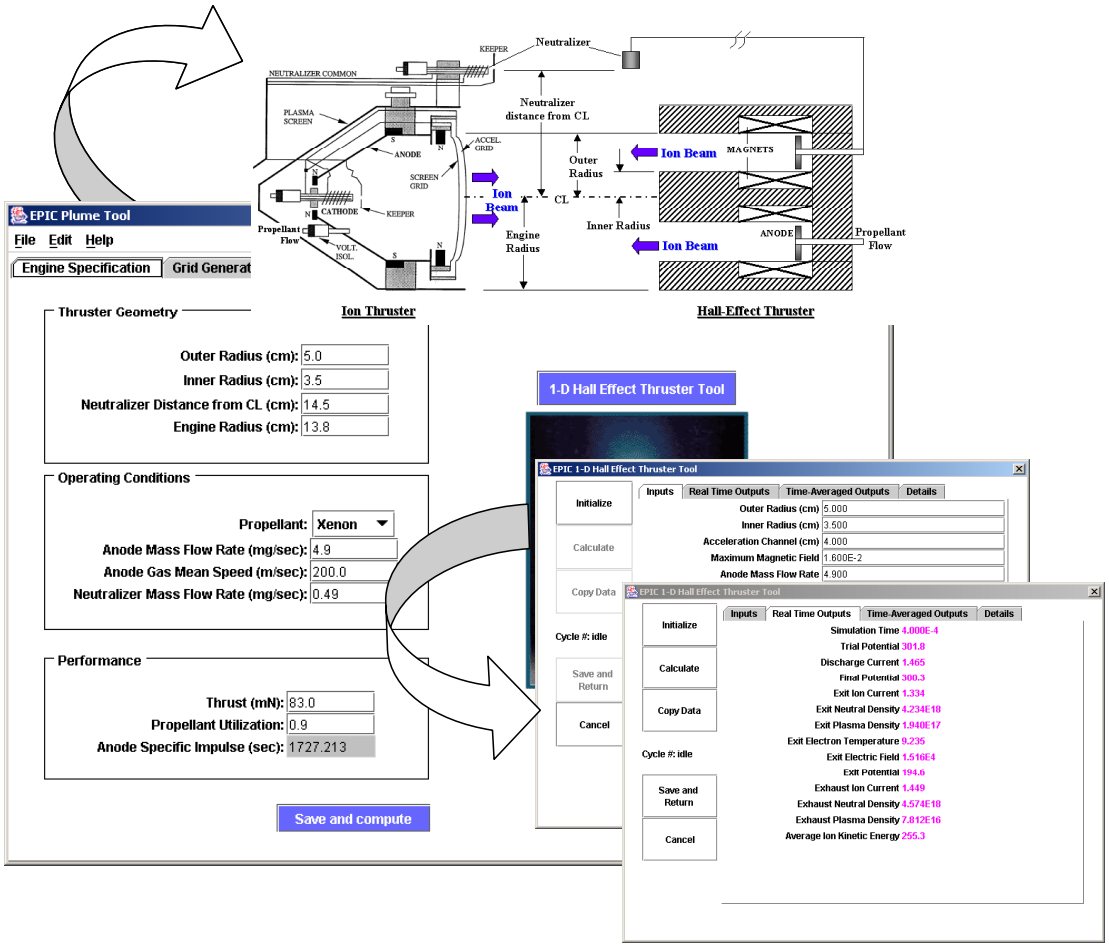

Figure 3. PlumeTool's Engine Specification window may be launched from the *EPIC* main interface.

Ions are accelerated by the electric field, $\vec{E} = -\nabla\phi$ according to,

$$m_i \frac{D\vec{v}_i}{Dt} = -e\nabla\phi \qquad (2)$$

where m_i and e are ion mass and electron charge, respectively. Since the drift velocity of the ions, $\vec{v}_i$, is much greater than their thermal velocity, the high velocity ions are modeled as a fluid. The steady state, conservation equations of mass and momentum are solved in 2-D (R-Z) geometry. The numerical algorithm for computing the expansion of the main ion beam is based on a Lagrangian approach in which discrete mass elements, or otherwise "macroparticles" are released from the thruster exit and are then "tracked" using fundamental trajectory kinematics.[10] The (normalized) current density profiles presently implemented along the exit of Ion and Hall thrusters are shown in Figure 4. For an Ion thruster the current density profile is assumed to vary parabolically with radius, *i.e.* as $[1-(r/r_{outET})^2]$, where r_{outET} is the thruster outer radius. The ion velocity components V_r and V_z are determined by assuming that all ions are emitted from a point

source behind the acceleration grid, with a maximum divergence angle of α (currently set at 20 deg). Thus,

$$\bar{u}_r \equiv \frac{V_r}{|\mathbf{V}|} = \frac{\tan\alpha}{\sqrt{\tan^2\alpha + \dfrac{1}{(r/r_{outET})^2}}}, \quad \bar{u}_z \equiv \frac{V_z}{|\mathbf{V}|} = \sqrt{1-\bar{u}_r^{\,2}}$$
$$V = \sqrt{\frac{2eE_b}{m_i}} \tag{3}$$

where E_b is the main beam energy. For Hall thrusters the current density and velocity profiles at the exit are based on extrapolation of integrated measurements from the Busek-Primex Hall thruster (BPT-4000).[5]

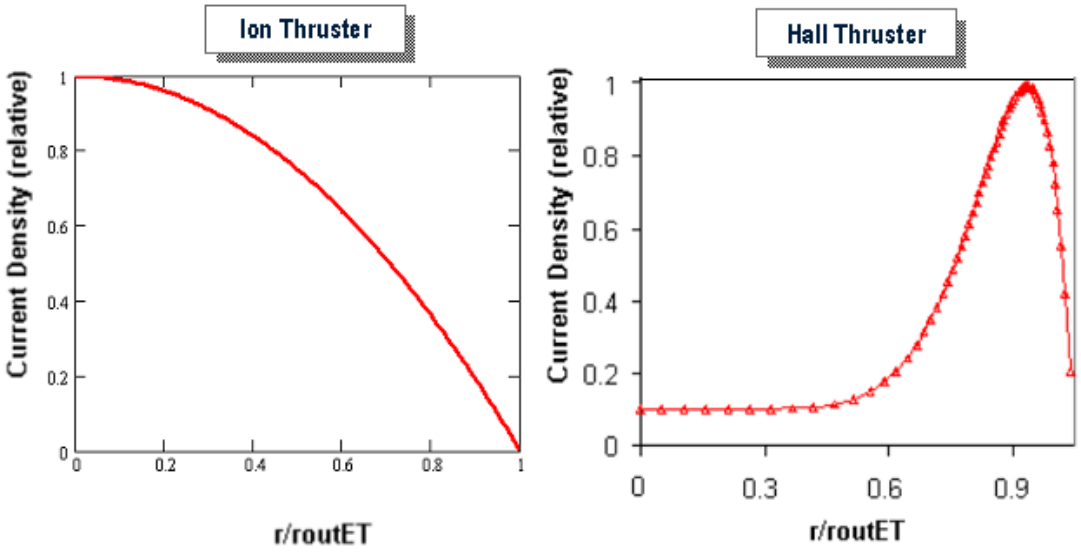

Figure 4. Normalized current density profiles for Ion (left) and Hall thrusters (right).

The neutral gas density in space is mainly due to particles from the thruster and from the hollow cathode. The beam of neutrals from the thruster is computed using an annular anode gas flow model with isotropic emission from a ring. The profile of neutrals from the thruster is computed using two disk emissions defined by the solid angles subtended by each disk, and subtracting the smaller from the larger.[8] The flow rate of neutrals from the thruster exit is estimated using the known anode flow rate and propellant utilization. The hollow cathode is offset by a distance r_{HC} from the thruster. Its axial location is assumed to be at z=0. The constant-temperature neutrals are emitted isotropically from the neutralizer, and their speed is estimated based on the same assumptions used for the thruster neutrals.

Fast ions from the main beam undergo charge-exchange (CEX) with neutral particles, resulting in slow-moving ions and fast-moving neutrals,

$$_{fast}Xe^{+} + {}_{slow}Xe \rightarrow {}_{slow}Xe^{+} + {}_{fast}Xe \tag{4}$$

In *EPIC*'s 2-D plume code, charge-exchange is computed using a two-dimensional PIC method. The rate of CEX-ion production rate $\dot{n}_{CEX}$ is determined by,

$$\dot{n}_{CEX} = n\,\vec{v}_i \sigma_{CEX} n_o \tag{5}$$

In eqn (5) n_o is the particle density of neutrals and σ_{CEX} is the cross section associated with the charge exchange collision. In contrast to the approach for the calculation of the main beam ions, where direct use of equation (1) provides the potential, the PIC algorithm solves the 2-D Poisson's equation on a finite element grid and iterates until steady state CEX densities and potentials are consistent. The plasma density in Poisson's equation is the sum of the main-beam and CEX densities. The first (computed by the Lagrangian method), and the prescribed neutral gas profile (above), are used as input for the calculation. The computed (total) ion particle density in Figure 5 shows clearly the CEX ion regions in the plume of an SPT-100.

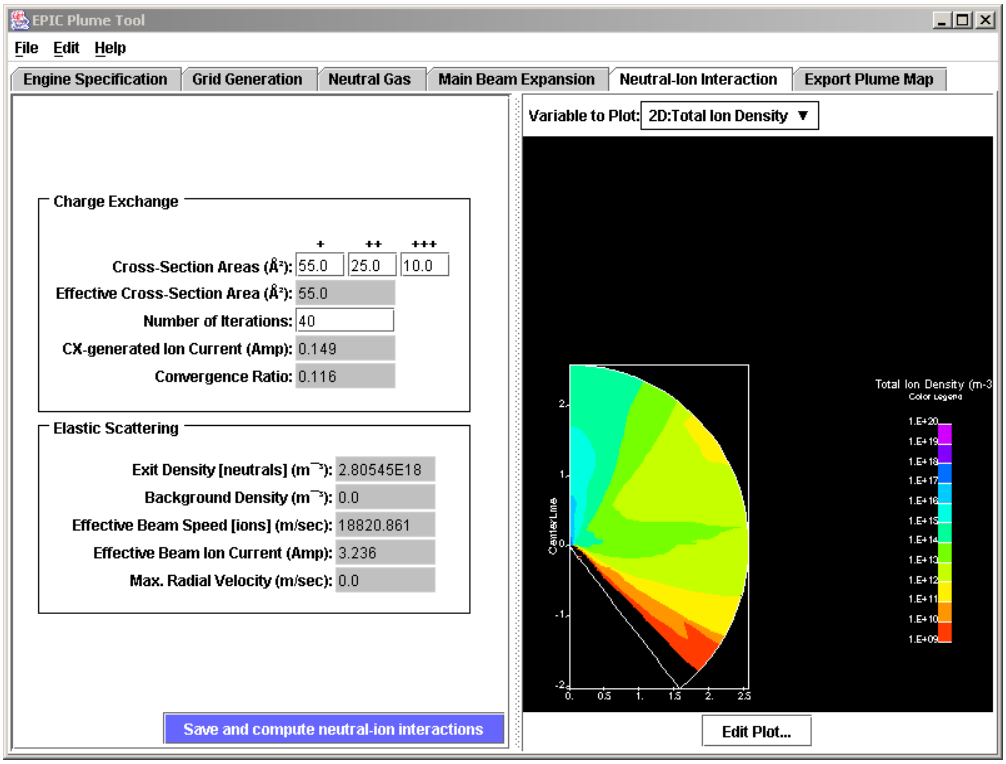

Figure 5. Total ion particle density in the plume of an SPT-100 as computed by PlumeTool's 2-D code.

With the revival of nuclear sources in space flight many EP systems are projected to operate at high power levels (>10 kW) for increased specific impulse and thrust. Main beam ion energies in such missions can range in the thousands of electron-volts (>5000 eV) with ion flow rates exceeding a few milligrams per second (>3 mg/sec). In these propulsion systems the need to quantify the effects of high-energy ions (>300 eV) that are produced by elastic scattering

between main beam ions and neutral particles becomes critical. The main reason is that these ions can be scattered to angles greater than those associated with the divergence of the main beam (which usually does not exceed 45 deg) and can therefore pose significant damage to surrounding spacecraft structures and diagnostics. An algorithm that determines the flux and energy of elastically scattered ions is part of PlumeTool; we outline here the basic approach.

In the classical sense, the differential cross section associated with an atomic collision can be calculated from the deflection angle, $\Theta(b,E_c,)$[11]

$$\Theta(b, E_c) = \pi - 2b\int_{r_m}^{\infty} \frac{dr}{r^2\sqrt{1-(b/r)^2-(V(r)/E_c)}} \tag{6}$$

where E_c is the center-of-mass collision energy, b is the impact parameter, r is the interatomic distance, r_m is the *classical turning point* (point of nearest approach), and V(r) is the interatomic potential energy. In the center of mass frame, θ, is given by $\theta = |\Theta|, \quad 0 < \theta < \pi$.

The differential cross section is obtained from equation (7) as follows:

$$I(\theta, E_c) \equiv \frac{\partial\,\sigma}{\partial\,\Omega_{CM}} = \left|\frac{b}{(d\theta/db)\sin\theta}\right| \tag{7}$$

where σ is the collision cross section. The solid angle is given by $d\Omega=2\pi\sin\theta\,d\theta$. In the numerical model, given the angle and energy for each scattering pair (s-point to f-node as it will be shown later), the impact parameter is obtained by solving equation (6) using a Newton-Raphson method. The solution requires *apriori* knowledge of the interatomic potential function, V(r), which may in general have both attractive and repulsive contributions. At the high energies of interest in the present work (>300 eV) the deflection function barely exhibits a minimum as the scattering is almost solely governed by the repulsive part of the interaction potential. In such cases the classical approach (vs. the more rigorous quantum mechanical approach[12]) is sufficiently accurate. Currently, only the repulsive potential function is included in the model. The coefficients that define the functional form of V(r) have been derived from averaged potentials by Amarouche, M., *et al.*[13] and are listed in Katz, I., *et al*.[14] The derivative in equation (7) is computed using a first order forward finite difference.

We refer to Figure 6 to describe the formulation used for the determination of the density, average velocity and average energy of the elastically scattered ions. It is assumed that at the scattering point "s", one scattering event occurs as the main beam ion flux, $F_b=n_{b,s}u_{b,s}$ impacts one stationary neutral particle. Once scattered, each ion does not undergo another collision. Then the particle flow rate dN/dt (#particles N per unit time t) through the incremental surface $dS=\rho^2 d\Omega$, is given by:

$$\frac{dN}{dt} = F_b I_{ion}\frac{dS}{\rho^2} \tag{8}$$

where dΩ is the solid angle subtended by dS at point "s", $I_{ion}(\theta) = \frac{1}{2} I(\theta) + \frac{1}{2} I(\pi - \theta)$ and ρ is the distance between "s" and field node "f". The particle flow rate is also equal to,

$$\frac{dN}{dt} = n_{sf} u_{sf} dS \tag{9}$$

where, n_{sf} and u_{sf} are the elastically scattered particle density and speed, respectively, at the field node "f". Combining equations (8) and (9) we obtain,

$$F_{sf} = n_{sf} u_{sf} = \frac{4 F_b I_{ion} \cos \vartheta_{sf}}{\rho^2} \tag{10}$$

Equation (10) expresses the magnitude of the flux vector, $F_{sf} = |\mathbf{F}_{sf}|$, at the field node location $f(r_f, z_f, 0)$ due to scattering of the main beam flux F_b by a neutral at the scattering point $s(r_s, z_s, \phi)$. We note the additional term $4\cos\vartheta_{sf}$ introduced in equation (10) by the transformation from the center-of-mass frame to the laboratory frame.[9]

It is also assumed that the distance between nodes is much larger than the mean free path for ion-neutral collisions. Since both momentum and kinetic energy are conserved in the scattering event we can write,

$$u_{sf}^{\ 2} = u_{b,s}^{\ 2} \cos^2 \vartheta_{sf} \tag{11}$$

which also yields the density of scattered ions at "f", n_{sf} as follows:

$$n_{sf} = \frac{F_{sf}}{u_{sf}} = \frac{4 n_b u_{b,s} I_{ion} \cos \vartheta_{sf}}{\rho^2 u_{b,s} \cos \vartheta_{sf}} = \frac{4 n_b I_{ion}}{\rho^2} \tag{12}$$

The contributions to "f" from all points "s" along the scattering ring defined by each scattering node at $s(r_s, z_s, 0)$, and by all nodes is summed as shown in equation (13),

$$n_f(r,z) = \int_V \frac{4 n_b I_{ion}}{\rho^2} n_o dV \tag{13}$$

where n_o is the neutral particle density at the scattering node.

Finally, we compute the average energy of the scattered ions at each field node using the energy flux and the particle flux as shown in equation (14) below. E_{sf} is computed using equation (11).

$$\overline{E}_f(r,z) = \frac{\int_V E_{sf} F_{sf} n_o dV}{\int_V F_{sf} n_o dV} \tag{14}$$

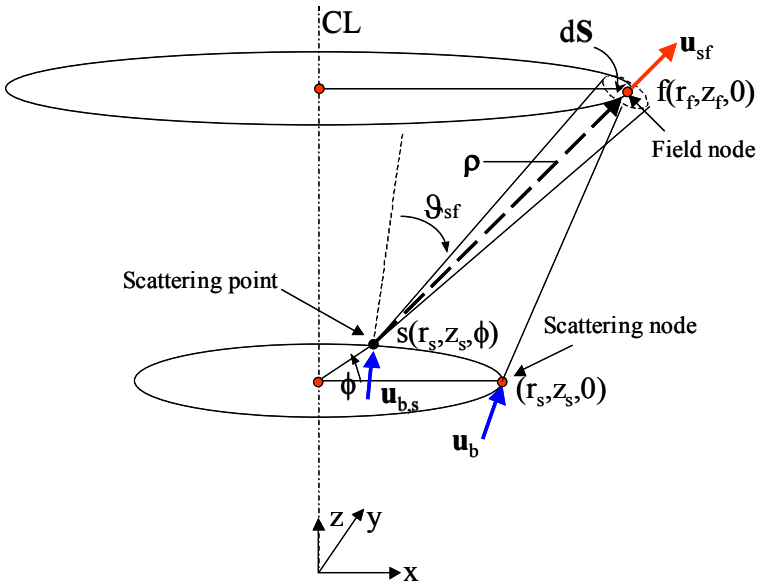

Figure 6. Elastic scattering geometry used to formulate the generalized algorithm in *EPIC*'s plume code. The four (red) corners define a portion of the plume computational plane.

Comparisons with idealized test problems and measurements have been conducted for the purpose of validating the algorithm. The comparisons showed good agreement and are described in Gardner, B.M., *et al.*,[10] and Mikellides, I.G., *et al.*[9] An example calculation for the BPT-4000 plume is shown in Figure 7 (the computed main beam only is shown on the left). The computed particle density of the elastically scattered ions, when the thruster was operated in the laboratory, is depicted in Figure 7, right. The value of background pressure for this plume calculation was taken to be 3.3e-5 Torr. Figure 8 shows the plume map for an SPT-100 imported in the main *EPIC* interface.

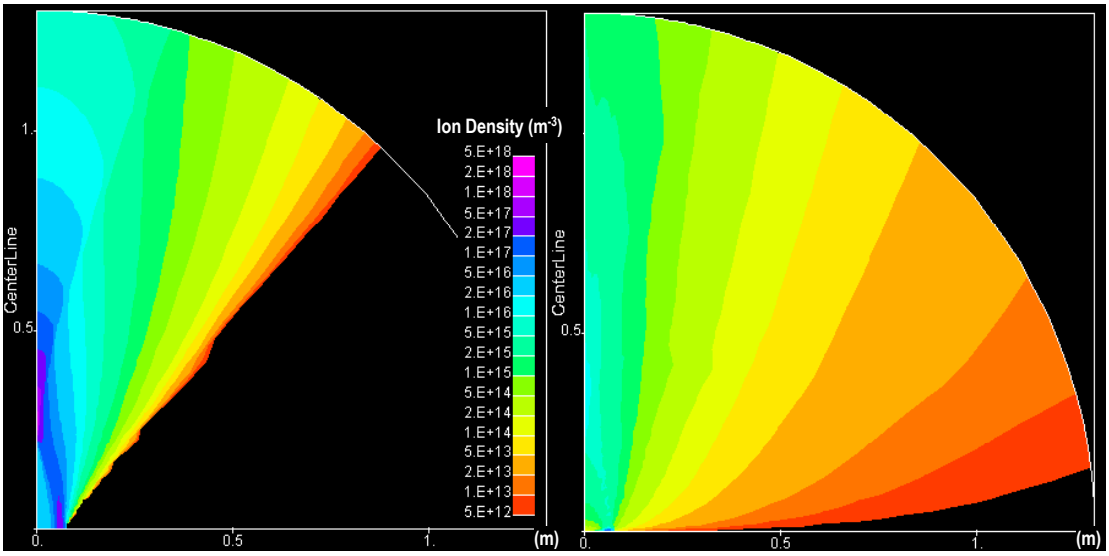

Figure 7. Computed main beam (left) and elastically-scattered ion density (right) (using the *EPIC* algorithm), in the plume of the BPT-4000 engine operating in the laboratory (P=3.3e-5 Torr).

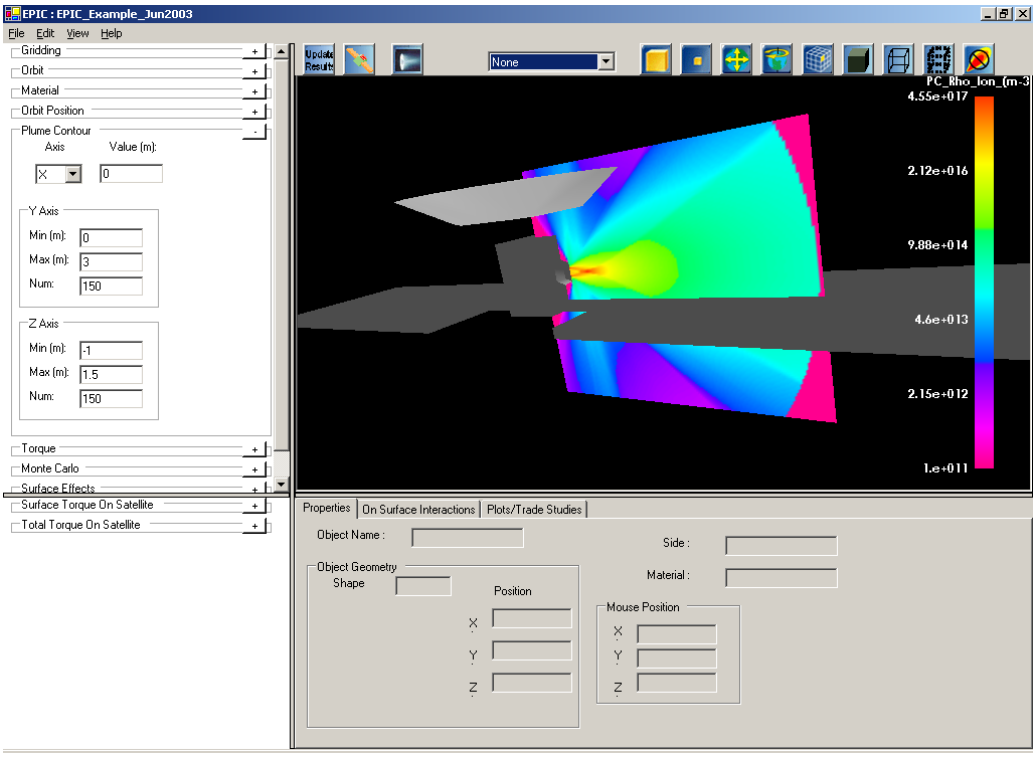

Figure 8. Imported SPT-100 plume showing ion particle density profile.

1-D HET Code. In electric propulsion systems such as HETs for example, it is difficult to predict conditions at the exit using empirical measurements exclusively. Furthermore, the effect of thruster geometry and operating conditions on the exhaust, and consequently the surrounding

S/C surfaces, is difficult to assess by simple extrapolation of measured quantities associated with a given thruster. Therefore, specifically for Hall thruster plumes *EPIC* incorporates a one-dimensional computer model of the acceleration region, which can be launched from within PlumeTool, as shown in Figure 3. The physics include collisions of electrons with neutrals, ions, and walls. The electron energy treatment includes ohmic heating, thermal diffusion and convection, and losses due to ionization, collisions with walls and atomic excitation.[15]

Plume-spacecraft interactions

The ability to assess a variety of spacecraft interactions with electric propulsion plumes, in three dimensions, is the bulk of *EPIC*'s capability. The interactions models presently allowable by *EPIC* have been described in greater detail elsewhere.[16] We summarize here the most commonly used interactions capabilities of *EPIC*. It is noted *apriori* that *EPIC* does not account for any plume perturbations caused by the presence of the spacecraft. Therefore, except for straight-line shadowing, all interactions calculations are carried out using a fixed plume profile (as provided by PlumeTool).

Fluxes to Surfaces. The ion flux F_{ijk} (atoms m^{-2} s^{-1}) at any point, i, on a surface j, due to plume component k, is calculated as follows:

$$F_{ijk} = \rho_{ik} \mathbf{v}_{ik} \cdot \mathbf{n}_j \tag{15}$$

where ρ_{ik} is the density at point "i" due to component "k", $\mathbf{v}_{ik}$ is the ion velocity (m/s) at point "i" of component "k", and $\mathbf{n}_j$ is the outward normal to surface "j". Fluxes to points on surfaces account for the interference ("blocking") by other spacecraft surfaces. Specifically, if a straight line between the point in question and the thruster orifice intercepts any other surface the flux is zero.

Surface Sputtering. The instantaneous sputtering rate (atoms m^{-2} s^{-1}) of a spacecraft surface "j" at a point "i" due to thruster plume impingement, R^S_{ij}, is

$$R^S_{ij} = \sum_k Y_{ijk} F_{ijk} \tag{16}$$

where "k" is summed over the plume components and Y_{ijk} is the sputter yield at point "i" of surface "j" from plume component "k". The sputter yield depends on the material, the energy of the ions impacting the surface and the angle between the flux vector and the surface-normal. All formulae are fits to sputtering measurements using the following functional form:

$$Y(E, \theta) = (a+bE)(1.00-0.72\ \theta +11.72\ \theta^{2} -3.13\ \theta^{3} -2.57\ \theta^{4}) \tag{17}$$

The yield is given in atoms/ion. The angular dependence is from Roussel *et. al*.[17]

Erosion/Deposition. The erosion/deposition rate at point "i" of surface "j" is the difference between the deposition and erosion rates.

$$R_{ij} = R_{ij}^{D} - R_{ij}^{S} \tag{18}$$

If $R_{ij} > 0$ it is a net deposition rate; if $R_{ij} < 0$, it is a net erosion rate.

The deposition rate is given by

$$R_{ij}^{D} = \sum_n \left(R_n^{S} \cos\left(\theta_{ijn}\right) \frac{\Omega_{in}}{2\pi} \right) \tag{19}$$

where R_n^S is average sputtering rate of all the points of surface "n," Ω_{in} is the solid angle subtended by surface "n" viewed from point "i" and θ_{ijn} is the angle between the normal of surface "j" and a ray from the centroid of surface "n" to the point "i".

The thickness or depth is calculated from the total particle deposition/erosion and the material "Density (gm cm^{-3})" and "Molecule mass" of the underlying material. Material properties are displayed on the "Material" sub-panel. The net erosion/deposition rate is computed at a randomly chosen selection of points over the mission (as specified in the "Monte Carlo" sub-panel) and then averaged. The mission averaged value is the product of the mission averaged rate and the "Mission Duration (days)" shown in the Orbit sub-panel. This ensures that time-dependent changes in spacecraft geometry (such as solar array sun-pointing) are included in the calculation.

For example, one may be interested in determining how long the thruster can be operated before material from the solar array is eroded below a critical thickness. The user can plot erosion of a specific element as a function of time of thruster operation as shown in Figure 9. The solar array element chosen is shown on the spacecraft. Additional variables may be plotted in this manner. The green line depicted on the plot of Figure 9 is material deposited on the antenna element that is highlighted in red in Figure 2.

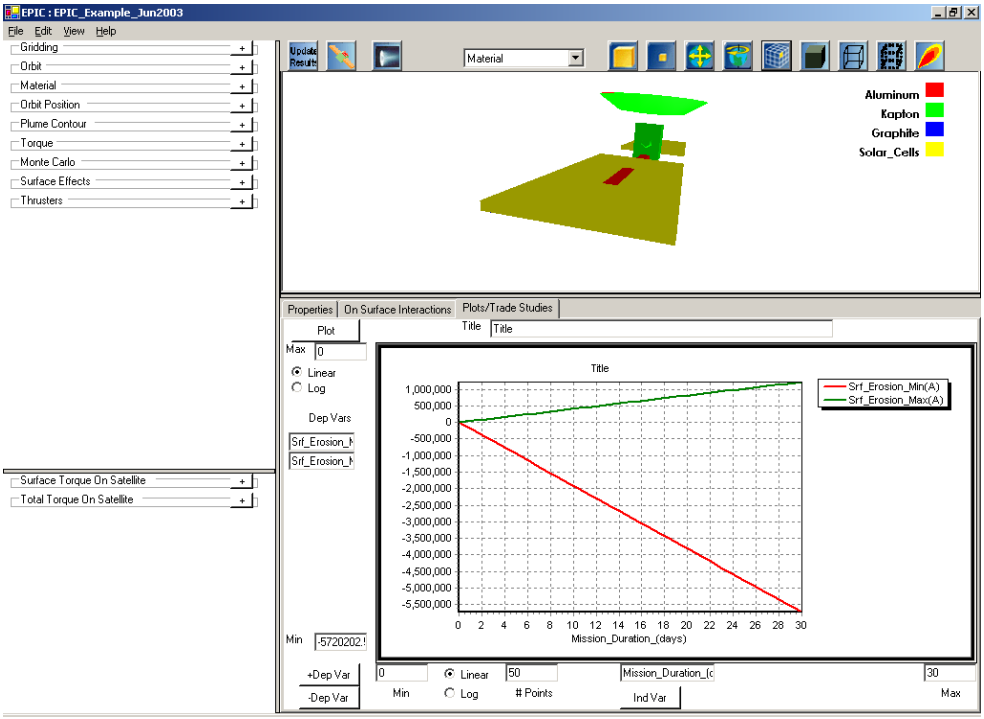

Figure 9. Trade studies may be performed using the Plots/Trade Studies tab. The plot shows erosion and deposition on critical spacecraft surfaces as a function of thruster operation time.

Induced Torques. Many electric thrusters produce torques that are comparable to or larger than those produced by gravity gradient, solar radiation, magnetic and aerodynamic effects, but are generally far smaller than torques resulting from chemical propulsion. Thrusters used for station-keeping or other purposes may produce unwanted torques, either directly or through the reflection of plume ions and neutrals from spacecraft surfaces.

There are in general two limiting cases for the reflection of plume particles from surfaces. In *specular* elastic reflection the component of momentum normal to the surface is reversed. Reflection from rough surfaces leads to *diffuse* reflection in which the outgoing particle travels in a random direction, *i.e*., particles are reflected in all directions regardless of the angle of incidence. In the case shown in Figure 10 (torque results are shown on the bottom left), specular reflection has been chosen. The general *EPIC* model used to determine the induced torques on the spacecraft during thruster operation accounts for contributions from the thrust and from the impingement of the exhaust on surfaces:

$$\boldsymbol{\Gamma} = \sum_{T} \Delta \mathbf{R}_T \times (-\mathbf{f}_T) + \sum_{j} \Delta \mathbf{R}_j \times \mathbf{f}_j \tag{20}$$

where, $\mathbf{f}_j$ is the force imparted onto a surface j from plume particles, $\mathbf{f}_T$ is the thrust vector, $\boldsymbol{\Gamma}$ is the torque, $\Delta \mathbf{R}_T$ is the position vector of a thruster from a reference point and $\Delta \mathbf{R}_j$ is the position vector of a surface j from a reference point. In the model, the contribution from plume

impact is computed using one of the two limiting cases mentioned above: *specular* (elastic) reflection from the surface and *fully diffuse* reflection.

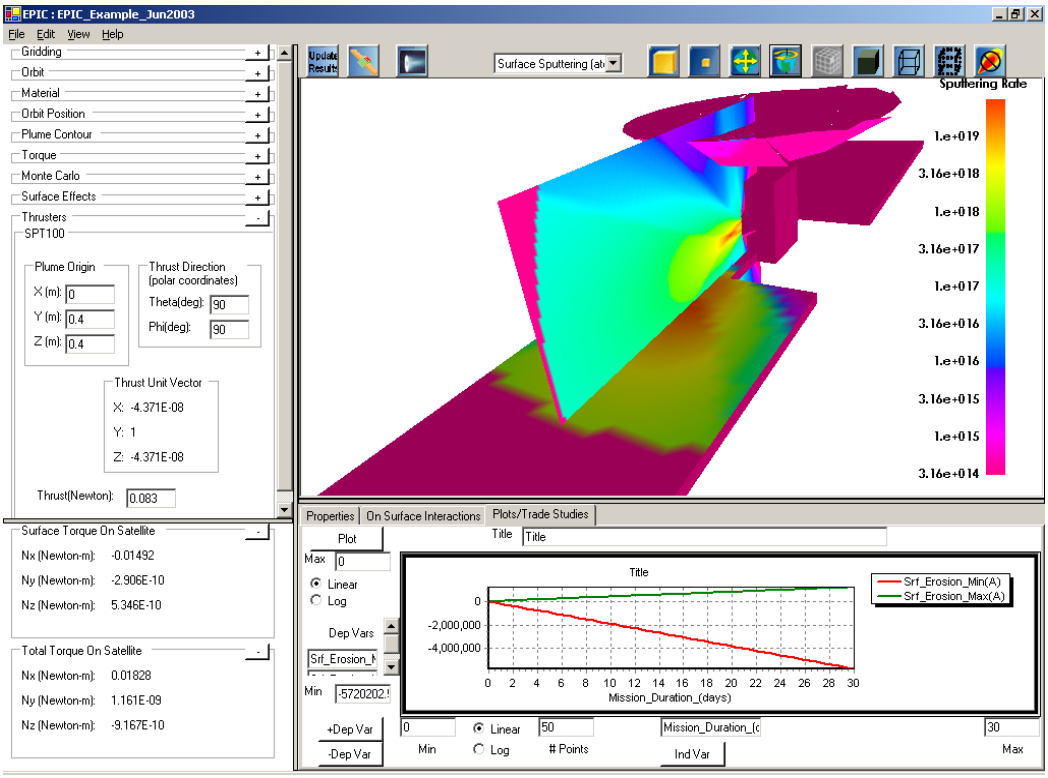

Figure 10. Additional calculations and display options in *EPIC*.

Summary

The near and long-term benefits of electric propulsion are now widely recognized and accepted by both the government and the private sector. Solar-powered EP systems have been used successfully in numerous missions, with many more readied for near-term applications. Through Project Prometheus NASA now also focuses on nuclear electric propulsion (NEP) as the "way to defeat distance and time in exploring the outer planets." However, integration of EP systems on spacecraft presents serious barriers to implementation. In the absence of accurate assessments, interactions between the propulsion system and the surrounding spacecraft components may threaten mission success. Quantifying such interactions within the design of specific spacecraft is a complex task with no comprehensive tools for the spacecraft engineer in existence. In response to the growing need by the electric propulsion community for a stand-alone design tool that addresses these issues, SAIC is developing the Electric Propulsion Interactions Code, *EPIC*. *EPIC* is a windows-based interactive, computer tool that allows the spacecraft designer to conduct the complete interactions process including geometry definition, EP plume generation, and interactions evaluation in 3-D. A fully functional beta version of the tool was delivered to the SEE Program on July, 2003.

Acknowledgments

This work is supported by the Space Environments and Effects (SEE) Program at the NASA Marshall Space Flight Center under contract no. NAS 8-02-028.

References

1. Mikellides. I.G., Kuharski, R.A., Mandell, M.J., Gardner, B.M., "Assessment of Spacecraft Systems Integration Using The Electric Propulsion Interactions Code (EPIC)," AIAA Paper 2002-3667, July 2002.

2. Fife, J.M., et al., "The Development of a Flexible, Usable Plasma Interaction Modeling System," AIAA Paper 2002-4267, July 2002.

3. Purrin, V., Pogarieloff, D., Metois, P., and Brosse, S., "PPS Effects on Satellites Analyses and Tools in Alcatel Space Industries," IEPC Paper 01-262, Oct. 2001.

4. Davis, V.A., et al., "Ion Engine Generated Charge Exchange Environment: Comparison Between NSTAR Flight Data and Numerical Simulations," AIAA Paper 2000-3529, July 2000.

5. Gardner, B.M., Katz, I., Davis, A. and Mandell, M. "Hall Current Thruster (HCT) IR&D Plume Environment Characterization and Electrostatic Return Current Assessment" Final Report for Lockheed Martin Space Systems Company (MTSD-DPR-01-16648), Jan. 2001.

6. Mandell, M.J., et al., "Nascap-2k, A Spacecraft Charging Analyis Code for the 21st Century," AIAA Paper 2001-0957, Jan. 2001.

7. Mikellides. I.G., et al., "Assessment of High-Voltage Solar Array Concepts for a Direct Drive Hall Effect Thruster System," AIAA Paper 2003-4725, July 2003.

8. Mikellides, I.G., et al., "Plume Modeling of Stationary Plasma Thrusters and Interactions with the Express-A Spacecraft," J. of Spacecraft and Rockets, Vol. 39, No. 6, 2002, pp. 894-903.

9. Mikellides. I.G., et al., "The Electric Propulsion Interactions Code (EPIC): A Member of the NASA Space Environment and Effects Program Toolset," AIAA Paper 2003-4871, July 2003.

10. Gardner, B.M., et al., "Electric Propulsion Interactions Code, EPIC," Final Report (No. SAIC 02/2043) prepared under NASA Contract #NAS8 02028, Nov. 2002.

11. Child, M., Molecular Collision Theory, Academic Press, London, 1974, pp. 9-20.

12. Mott, N.F., and Massey, H.S.W., The Theory of Atomic Collisions, Ch. I,V and XII, Clarendon Press, Oxford, 1949.

13. Amarouche, M., Durand, G. and Marlieu, J.P., "Structure and Stability of Xe_n^+ Clusters," J. Chem. Physics, Vol. 88, 1988, pp. 1010-1018.

14. Katz, I., et al., "A Hall Effect Thruster Plume Model Including Large-Angle Elastic Scattering," AIAA Paper 2001-3355, July 2001.

15. Mikellides, I.G., Katz, I., Mandell, M.J., "Modeling the Exit Plane of a Hall Effect Thruster," AIAA Paper 01-3505, July 2001.

16. Mikellides, I.G., et al., "A Hall-Effect Thruster Plume and Spacecraft Interactions Modeling Package," IEPC Paper 01-251, Oct. 2001.

17. Roussel, J.F., Bernard, J., Garnier, Y., "Numerical Simulation of Induced Environment, Sputtering and Contamination of Satellite due to Electric Propulsion," ONERA-CERT/DERTS, Proc. Second European Spacecraft Propulsion Conference, May 1997.