Multiscale Modeling of Hall Thrusters

Lubos Brieda* and Michael Keidar†

The George Washington University, Washington, D.C. 20052

In this paper, we introduce methodology for performing self-consistent multiscale modeling of Hall thrusters and plume interactions. Our approach allows fast calculation of transport properties using a desktop workstation. We use a one-dimensional kinetic code to estimate mobility along several magnetic field lines. This mobility is used to correct the analytical classic plus Bohm mobility in an axisymmetric Hall thruster code. Ion velocities at the thruster exit plane are then sampled and used as the source model for a three-dimensional plume/spacecraft interaction code. Details of the three models are presented, and analysis is performed for a cluster of 2.6cm Princeton cylindrical Hall thrusters operating on a generic spacecraft.

I. Introduction

Despite Hall thrusters having over 40 years of flight heritage (the first variant, SPT-50, was flown aboard the Soviet Meteor spacecraft in 1971), the community still lacks a tool capable of predictively modeling these devices. The closest to an industry standard, COTS-level simulation program is HPHall. This 2D axisymmetric code was originally developed at MIT by Fife,[1] and was subsequently improved by others.[2,3] Other researchers have developed their own codes based on the basic premise of HPHall.[4–7] These code share a common approach. Ions are treated as kinetic particles while electrons are modeled as a quasi-1D fluid. Quasineutrality is assumed, and the kinetically determined plasma density is used to update the electron temperature in direction perpendicular to the magnetic field lines. Potential is recovered by considering current conservation and assuming electron thermalization along field lines. This model is generally valid - in Hall thrusters electrons are magnetized, and thus their motion can be decoupled into two distinct modes: the unhindered motion along the field line, and the mobility-driven diffusion across them. The mobility term comes into play in the electron momentum equation used to determine the cross-field velocity. This is one of the terms in the energy equation. HPHall relies on the classical $1/B^2$ collision driven transport combined with an anomalous $1/B$ Bohm term. Mathematically, the mobility model is $\mu_{e,\hat{n}} = \mu_e/(1+\beta^2) + K_b/(16B)$, where μ_e is the mobility $\mu_e = e/\nu_{en} m_e$ for unmagnetized electrons, K_b is the user-determined Bohm parameter and $\beta = \omega_c/\nu$ is the Hall parameter. This formulation is not without problems. Transport is inherently a kinetic phenomenon. It is influenced to a large degree by wall collisions, secondary electron emissions, and temporal field variations. These are kinetic effects that the simple analytical model fails to capture. In addition, as pointed out previously by Sydorenko,[8] electron temperature in Hall thrusters is anisotropic, further complicating the model.

Hence, several other researchers have taken path completely opposite of HPHall, and have developed fully kinetic Hall thruster codes. Example of one such code is the work of Hirakawa.[9] Many other researchers have also contributed.[10–12] These fully kinetic codes have their own drawbacks. First, they require extremely small simulation time steps. Resolving the cyclotron rotation requires timesteps some six orders of magnitude smaller than required to push ions in HPHall. Second, since potential in these codes is obtained from solution of the Poisson's equation, the computational cells must be small enough to resolve the Debye length. These two requirements make fully kinetic approach a computationally daunting task for all but the smallest of thrusters. Even if tricks are played with relative electron-to-ion mass and the free space permittivity, these codes still require supercomputers and many days or weeks of computational effort.

Fully kinetic codes are useful tools for understanding basic physical processes occurring in Hall thrusters. They are not particularly useful, however, to a designer working on optimizing one of these devices. Designer

*PhD Candidate, Department of Mechanical and Aerospace Engineering, lubos.brieda@particleincell.com

†Associate Professor, Department of Mechanical and Aerospace Engineering, keidar@gwu.edu

at some aerospace corporation will likely not have access to supercomputing facilities, nor have the time to perform a trade study if each cases requires substantial computational effort. For this reason, we started working on a new multiscale approach to modeling Hall thrusters. The objective of our work is to develop a tool capable of self-consistently determining electron mobility and the thruster plasma properties of interest, but do it in such a way that a solution can be obtained on a standard desktop workstation in less than a day. To accomplish this goal, we divide the problem into the following three spatial scales:

Magnetic Field Line: On the spatial scale of a magnetic field lines, dynamics is driven by the cyclotron motion of electrons. Electrons are magnetized, and individual field lines can be considered independent of each other. Heavy particle and properties normal to the field lines are assumed frozen. This approach allows us to rapidly simulate electrons and recover mobility self-consistently. Leveraging modern multi-core architectures via multithreading allows us to study multiple field lines simultaneously.

Thruster Channel: On the spatial scale of the thruster, plasma is assumed to be quasineutral, and electron density can be obtained from kinetic ions. Electron temperature and plasma potential is obtained by solving the quasi-1D equations.This approach is identical to HPHall, except that our method relies on the mobility determined by the kinetic analysis. Ions exiting the thruster at steady state are sampled to obtain a discretized source term for plume modeling.

Plume Environment: Outside the thruster exit, the magnetic field plays a negligible role. The plume is quasineutral except in low density sheath regions around the spacecraft. Of interest here is the formation of charge exchange ions and their impact on spacecraft components. Electron density is obtained from Boltzmann relationship, and potential can be solved by direct inversion or by solving Poisson's equation.

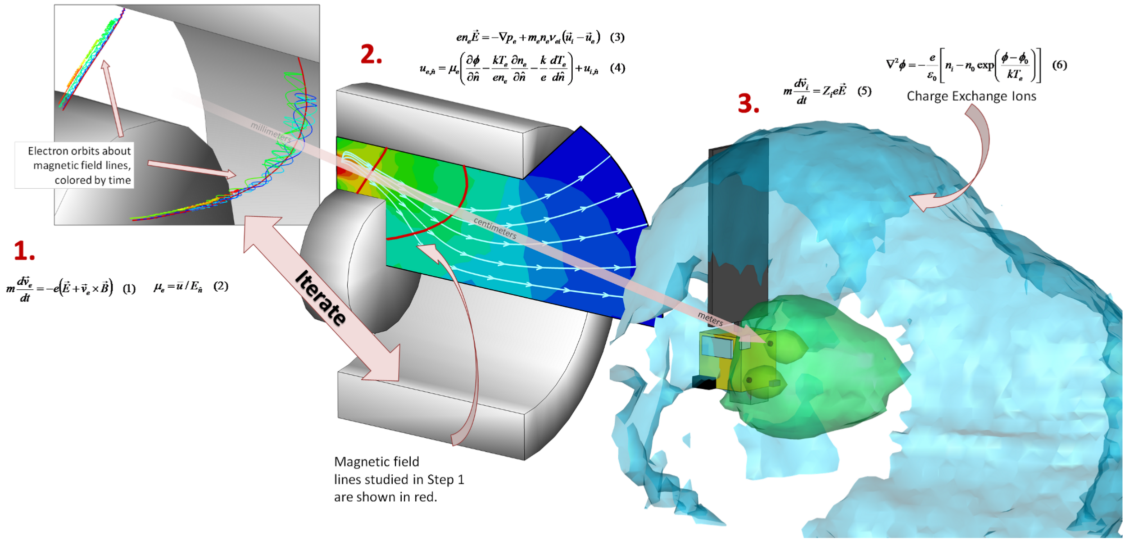

1.
$m\frac{d\vec{v}_e}{dt} = -e\left(\vec{E} + \vec{v}_e \times \vec{B}\right)$ (1)
$\mu_e = \bar{u} / E_{\hat{n}}$ (2)
Electron orbits about magnetic field lines, colored by time
millimeters
Iterate
2.
$en_e\vec{E} = -\nabla p_e + m_e n_e \nu_{ei}\left(\vec{u}_i - \vec{u}_e\right)$ (3)
$u_{e,\hat{n}} = \mu_e\left(\frac{\partial\phi}{\partial\hat{n}} - \frac{kT_e}{en_e}\frac{\partial n_e}{\partial\hat{n}} - \frac{k}{e}\frac{dT_e}{d\hat{n}}\right) + u_{i,\hat{n}}$ (4)
centimeters
Magnetic field lines studied in Step 1 are shown in red.
3.
$m\frac{d\vec{v}_i}{dt} = Z_i e\vec{E}$ (5)
$\nabla^2\phi = -\frac{e}{\varepsilon_0}\left[n_i - n_0\exp\left(\frac{\phi - \phi_0}{kT_e}\right)\right]$ (6)
Charge Exchange Ions
meters

Figure 1. Schematic of our multiscale approach. Codes 1. and 2. are designed to iterate until convergence. We then sample ions crossing the thruster exit plane to obtain the source model for plume analysis.

This approach is shown schematically in Figure 1. This formulation naturally lends itself to three codes, each concentrating on the physics of the respective spatial scale. The first code, *Lynx*, simulates the cyclotron motion of electrons about series of magnetic field lines. The second code, a general 2D axisymmetric/Cartesian plasma simulation tool *Starfish*, is still in development, and hence in this paper we utilized HPHall. The final piece is the 3D plume code *Draco*. We begin the paper by describing the thruster to which this approach was applied. We next skip to the second step, and use HPHall to obtain the initial plasma parameters which will serve as inputs to Lynx. We then perform a kinetic analysis to compute new values mobility. These are then used to update the HPHall solution and extract the plume source model. Repetitive iteration between steps 1 and 2 is left as a future item that will be attempted once Starfish development is complete. The paper concludes with a summary and a discussion of future work.

II. Thruster Details

We deploy our model to the 2.6cm Princeton Cylindrical Hall thruster (CHT). This thruster is described in greater detail in Ref. 13. Here we summarize just the parameters important to our analysis. The most important characteristic of this device is its non-standard geometry. While typical Hall thrusters consist of an annular channel, the CHT contains an annular upstream zone and a cylindrical acceleration zone. The lack of the inner wall in the acceleration region is expected to lead to increased thruster lifetime and improved performance due to reduced losses of ions to the walls. From the academic standpoint, this configuration also introduces interesting new physics. The magnetic field lines converge near the innerpole, resulting in a region of increased magnetic pressure. Electrons are then expected to be preferentially scattered to the outer wall, possibly resulting in a non-symmetric sheath. This finding was touched upon in our previous work.[14]

The walls of this SPT-type thruster are made of a dielectric material. The dielectric walls distinguish this thruster design from another Hall thruster variant, TAL (thruster with anode layer), in which the walls are conductive. One theory of SPT operation suggests that electron impacts of the dielectric material result in emission of secondary electrons from the material matrix. Since these electrons are not initially magnetized, they are free to migrate towards the anode. This theory is called near wall conductivity (NWC) and it's a feature that is captured by our kinetic code. These electrons are also significantly colder than the impacting particles, resulting in the cooling of the primary population.

Experimental measurements of this thruster have been presented in Refs. 15 and 16. The CHT can be operated in two modes based on the current applied to the magnetic coils. Our work correlates to the "direct" mode. In this configuration, magnetic field lines cross the channel between the outer wall and the innnerpole without forming a cusp near the outer wall. Electron temperature in the thruster was found to peak at 25 eV just outside the exit plane. Potential decays slowly through majority of the channel. Only 50V of potential drop were measured in the first 1.5 cm from the anode. Most of the potential drop occurs near the exit plane, in the acceleration zone. Additional 50V potential drop was measured to occur outside the thruster. Anode current was approximately 0.3A, and plasma density was $\sim 6 \times 10^{17}$ m^3.

III. Initial Results for Thruster Discharge

We used the HPHall simulation code to model the discharge channel. Schematic of our setup is shown in Figure 2. HPHall uses a structured but non-uniform mesh, boundary of which is shown in the figure. Such a mesh simplifies capturing the geometry of non-rectangular devices, as well as the downstream near-plume region. Internally, HPHall establishes another virtual mesh that is used to solve the electron equations. This mesh is illustrated in Figure 2. The vertical lines correspond to approximately equidistantly spaced magnetic field λ lines. It should be noted that this graphics does not completely correlate to the implementation in HPHall. The data structure used internally by the code uses a variable number of radial segments, which leads to a more complicated visualization problem. The mesh shown here, and in our subsequent analysis, used a uniform number of radial segments corresponding to the average number used by HPHall.

Of importance are the left and the right boundaries. These correspond to the anode and the cathode, respectively. Constant potential is applied upstream of the anode. The anode potential is computed by the code self consistently. The right boundary is the cathode line on which potential, density, and temperature are specified. Simple linear interpolation for temperature and potential is used downstream of the cathode. As can be seen from the figure, this downstream region can encompass a substantial fraction of the simulation domain. Important part of setting up an HPHall simulation is determining where to place the anode and the cathode lines. Typically, trade study is performed and the solution in the best agreement with experiments is selected.

HPHall injects kinetic electrons at the anode. Ionization model is then used to create ion particles. Ion positions are integrated using the particle in cell method. The code assumes charge neutrality, and hence $n_e = n_i + 2n_{i2}$, summation of singly and doubly charged ion populations. In order for quasineutrality to hold, electron temperatures must adjust accordingly to accommodate diffusion. HPHall solves the energy equation

$$\frac{\partial}{\partial t}\left(\frac{3}{2}n_e kT_e\right) + \nabla\cdot\left(\frac{3}{2}n_e kT_e\vec{u}_e + \vec{q}_e\right) + \nabla\cdot\left(n_e kT_e\vec{u}_e\right) = S_h - S_i \tag{1}$$

on the previously described lambda mesh. Constant electron temperature at each field line is assumed. This then allows each field line to be treaded as single volume element. Relevant properties, such as mobilities or

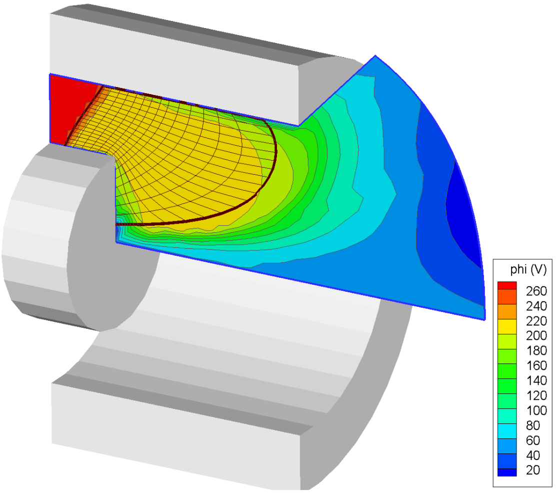


Figure 2. Simulation model of the cylindrical Hall thruster. Slice shows the HPHall computational domain. The mesh corresponds to the region in which the electron conservation equations are solved.

electron densities, are integrated along the line following the standard finite volume formulation. In other words, the computation is performed in a quasi-1D dimension. The conservation equations are solved only in the direction normal to the field lines, but 2D radial contribution is used to compute the coefficients at each point. Electron velocity, given by the momentum balance,

$$\vec{u}_e = \mu_e \left(\frac{d\phi^*}{d\hat{n}} + \frac{k}{e} \left(\ln(n_e) - 1 \right) \frac{dT_e}{d\hat{n}} \right) + u_i \tag{2}$$

is incorporated into the temperature solver. Here μ_e is the mobility term. Once temperature has been determined, thermalized potential ϕ^* can be computed at each lambda line from current conservation. The radial variation in potential is then recovered from the thermalized potential relationship, $\phi^* = \phi - kT_e/e \ln(n_e)$.

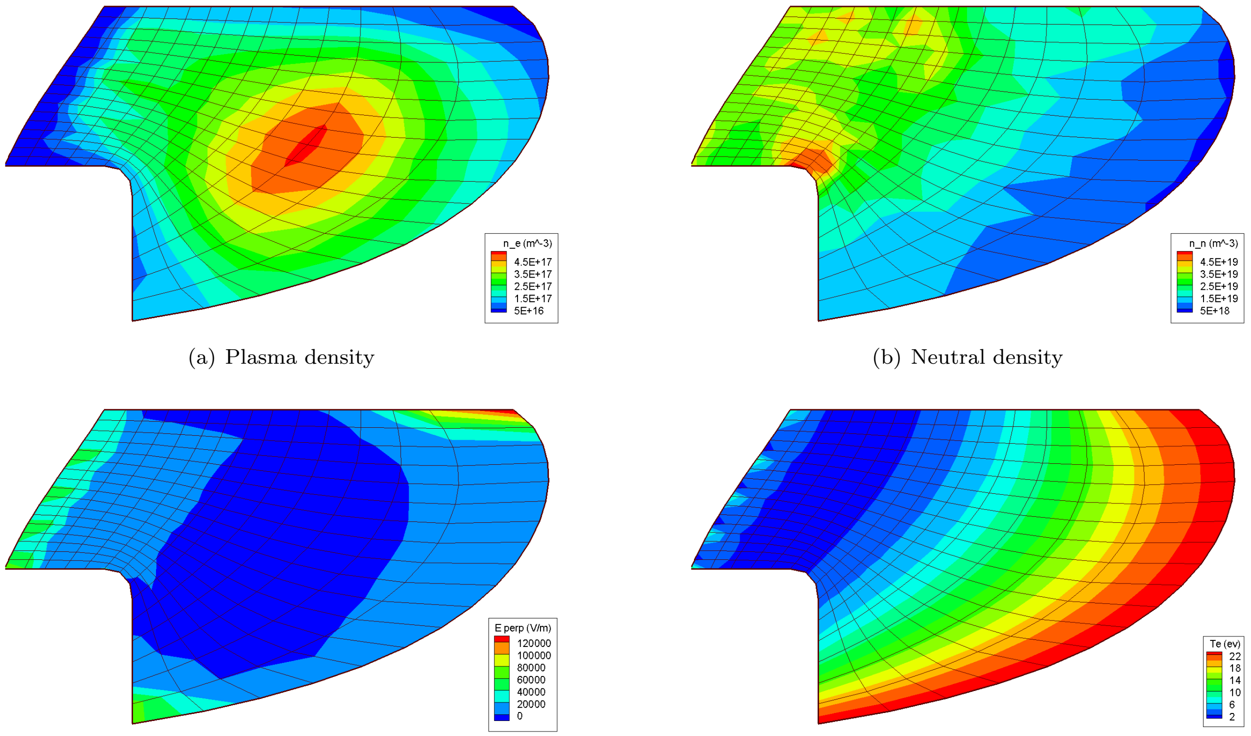
In our previous work[14] we performed a limited parametric study where we investigate several different cathode line positions and Bohm parameters with the goal of matching the experimental measurements for CHT. We were able to improve our correlation for this work. The relevant input settings are summarized in Table 1. Plasma properties along the thruster centerline are plotted in Figure 6. This figure shows both the results obtained using the analytical mobility model, and the results obtained subsequently using our kinetic method. The initial results with the analytical mobility are plotted with dashed lines. It should be noted that there are still discrepancies. For one, the HPHall-predicted potential distribution fails to capture the gradual rise to the anode potential. Instead, potential is actually seen to *decrease* upstream of the start of the acceleration zone. This indicates that fraction of ions born in this regions will have the tendency to flow towards the anode. One of the goals for this work was to determine if any improvement can be achieved by modifying the mobility distribution using the kinetic code.

A. Kinetic Code Inputs

The kinetic code requires as inputs information related to the global state of the discharge. We developed a simple code to contour HPHall results (`2d_ave_tp.dat`) along the lambda lines for computation. Contouring starts by searching for the corresponding value along the bottom edge of the computational domain. Edge cuts are then determined by linear interpolation of node values, and the cuts are connected to form a spline. Properties of interest are then interpolated onto the spline control points. Figure 3 shows the parameters. Along with the magnetic field profile (not shown here), these five parameters serve as inputs to the kinetic code. The kinetic code is described in the next section.

Table 1. Summary of critical inputs for the HPHall simulation

Parameter	Value
mass flow rate	4×10^{-7} (kg/s)
discharge voltage	275 V
anode line position	(0.0520, 0.010) m
cathode line position	(0.0670, 0.010) m
cathode temperature	26 eV
cathode potential	210.6 V
cathode density	$6{\times}10^{17}$ m^{-3}
cathode emitter potential	-20 V
Bohm coefficient	1.0



(a) Plasma density

(b) Neutral density

(c) Normal component of electric field

(d) Electron temperature

Figure 3. Plasma parameters serving as inputs to the kinetic code. Kinetic simulation is performed for each field line (vertical grid lines). Additional input, which is not shown here, is the magnetic field strength.

IV. Mobility Calculation with Lynx

The inputs in Figure 3 are next processed with a kinetic code called Lynx. Lynx is a kinetic code that self-consistently calculates the radial variation in electron mobility along a magnetic field line.[14,17] Lynx simulates only electrons. Ions and neutrals are assumed to remain frozen during the time necessary to compute electron trajectories. The heavy particles thus form a fixed background with which the electrons interact during their cyclotron motion about the field line. By repeating the calculation for all the lambda lines making up the HPHall electron mesh, we can obtain a two-dimensional variation in mobility. This kinetically determined mobility can then be used in HPHall in lieu of the analytical model.

Lynx is implemented in Java. Although historically Java performance was not competitive with languages such as C/C++, this is no longer the case. Modern Java compilers generate codes that perform at speeds comparable and in some cases even exceeding native C++ implementation.[18] In addition, Java offers a large standard library with support for data management, as well as GUI, networking, and graphics rendering. In addition, Java natively supports multithreading which allows us to take advantage of modern multi-core system architectures, and run Lynx concurrently for multiple field lines.[19] Similar functionality can be achieved in C++ through the Boost libraries. However, that step requires downloading, configuring, and compiling the massive libraries. In Java, such support is provided natively.

A. Execution and Topology

Lynx simulation commences with the code importing the 2D mesh shown previously in Figure 3. An additional settings file is also processed. This settings file specifies general simulation parameters such as wall temperature, number of electrons per field line, and the number of time steps. It also controls which diffusion-inducing processes, such as collisions or wall effects should be included. Lynx next instantiates a new simulation object for each magnetic field line. A simple scheduler was implemented to launch the threads and monitor them for completion. The number of threads running concurrently is limited by the number of available CPU cores.

The computational domain for each Lynx simulation thread is a one dimensional domain corresponding to a particular lambda line. In Figure 3, each vertical grid line corresponds to an individual simulation. The number of nodes along the line differed from the coarse mesh shown in this figure and was determined automatically such that $\Delta s = 0.5\lambda_D$. On average, 100 nodes were used per magnetic field line. Note, the coarse input mesh was used to store transport properties. The coarser mesh was selected in order to reduce statistical noise errors. Several fixed properties are set at each grid node during the initialization. These include n_i and n_0, ion and neutral densities, $E_\perp$, perpendicular component of electric field, B, magnetic field strength, and $\partial B/\partial s$, magnetic field gradient.

B. Particle Loading

The simulation then continues by loading the electron particles. Electrons are loaded at grid nodes, with the number of electrons obtained by multiplying the ion density by cell volume and scaling by the particle weight. As pointed out by Fox,[11] careful sampling of electrons is necessary in order to resolve the high energy tail of the velocity distribution function. We did not take this approach at this time, but plan to investigate the effect of variable particle weight on transport and near wall conductivity in the near future. The initial electron temperature at each field line is one of the inputs from HPHall. Isotropic temperature distribution was used for particle loading. Particles are loaded such that the axial position of the guiding center is $z = 0$. Initial number of particles was 50,000.

The initial plasma sheath profile needs to be determined next. Since ions are frozen, the establishment of equilibrium sheath is marked by zero electron current to the walls. During this initial period, collisions were not performed, and secondary electrons were not emitted. Instead, particles impacting the walls were simply removed form the simulation domain. These processes were ignored since we are interested in determining the electron density distribution arising from the thermal and potential balance. In our simulations, we found that steady state was achieved after approximately 1500 time steps. However, for an added margin of safety, we required to code to complete 3000 steps before the simulation continued. The additional time steps have no impact on the results since once the steady state is achieved, electrons become trapped in the potential well between the walls and are unable to reach the walls.

Simulation then moves to the normal operating mode, with collisions and SEE turned on. Particles translating away from the field line, or impacting the walls were removed from the simulation. The code reinjected new particles into the simulation to keep the number of electrons at the steady state level. Total of 20,000 time steps were simulated. Averaging of results began at time step 5000. The intermediate period between steady state and averaging was excluded from averaging to avoid any possible initial transient effects from influencing the results. The timestep Δt was selected such that electrons completed single orbit in 75 time steps.

C. Collisions and Near Wall Conductivity

Electron collisions were modeled using the Monte Carlo method. In this method, source particles are collided with a stationary target cloud. The collision probability is determined from the background density, n_0 and collision frequency is $P = 1 - \exp(-n_0 \sigma_0 g \Delta t)$. Here σ_0 is the total collision cross-section due to all processes. For particles undergoing collision, the collision process was picked randomly according to the ratio of σ_i/σ_0. Post-collision velocity was computed by first sampling a random target velocity and a random impact angle. We then calculated the post-collision velocity from conservation of energy.

To reduce statistical errors and improve performance, collisions were computed only once every 4 timesteps. Three types of collisions were considered: momentum-transfer (electron-atom), ionization (electron-atom), and excitation (electron-atom). Coulomb collisions were not included in the present work. At low electron temperatures, polarization collisions dominate the momentum-transfer interaction between electrons and atoms.[20] Cross-section for this process was obtained from the analytical model[20]

$$\sigma_{Xe} = \left(\frac{\phi \alpha_p q^2}{\epsilon_0 m} \right)^{1/2} \frac{1}{v} \tag{3}$$

where α_r is the polarizability of the atom. It is given by $\alpha_r = 27.66 a_0^3$, where a_0 is the Bohr radius.[21] Cross-section for electron-ion collisions was given by

$$\sigma_{Xe^+} = \frac{8}{\pi} b_0^2 \ln \Lambda \tag{4}$$

where $b_0 = q_1 q_2 / (2\pi\epsilon_0 m v^2)$ is the distance of closest approach. The inelastic process, ionization and excitation, cross-sections were computed using the polynomial fit of Szabo.[22] These collisions were modeled by reducing the energy of the impacting electron by the ionization or excitation energy, and scattering the electron through a random angle.

Electrons impacting the dielectric walls were reflected back to the domain, absorbed, or generated secondary electrons according to the model of Sydorenko.[23] The secondary electron yield was given by,

$$\gamma = \left(\frac{E_p}{35\text{eV}} \right)^{0.5} \tag{5}$$

The SEE yield is non-negligible for $kTe = 10$eV. The SEE electrons were assumed to come off the surface unmagnetized, and were generated at the wall with initial direction given by a random velocity vector. Impacting electron knocking off a secondary electron was assumed to be absorbed by the wall to retain charge neutrality, and was removed from the simulation. Although we have implemented a simple analytical model to take into account surface charging of dielectric walls, we did not utilize it at present. Instead, potential was fixed at both walls at 0V.

Figure 4 shows the typical field line quantities. This particular plot was generated for a magnetic field line close to the anode. We can see that in the bulk region, electrons clearly follow the ions. This charge neutrality is captured by the potential solution, leading to zero electric field in this bulk region. A clear sheath forms near the walls. Electron density decays faster, as expected. Electric field forms near the walls to repel low energy electrons. Wall potential drop is ∼5.7kTe.

D. Transport Calculation

Previous paragraphs described the methods used to simulate magnetized electrons bounded between two walls. In order to make the model useful to our multiscale approach, we need to extract mobility from the solution. Conceptually, determining mobility is a trivial task. Mobility could be obtained as $\mu_j = \sum u_z / E_\perp$,

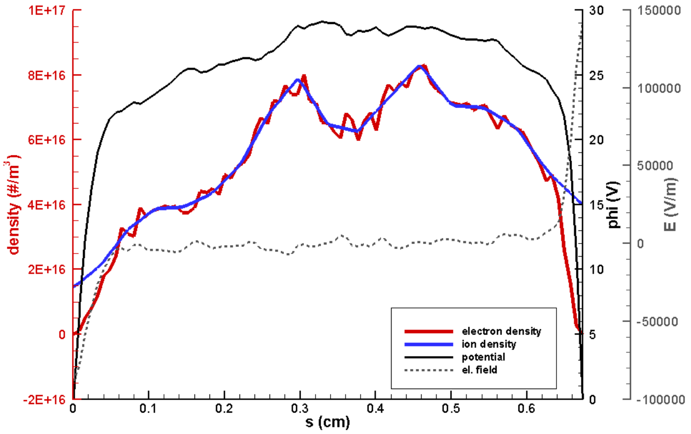


Figure 4. Plot of typical simulation results. Electrons are seen to closely follow the prescribed ion density, except in the sheath region, where they density decays more rapidly as expected. Plasma potential and tangential electric field profiles are also shown.

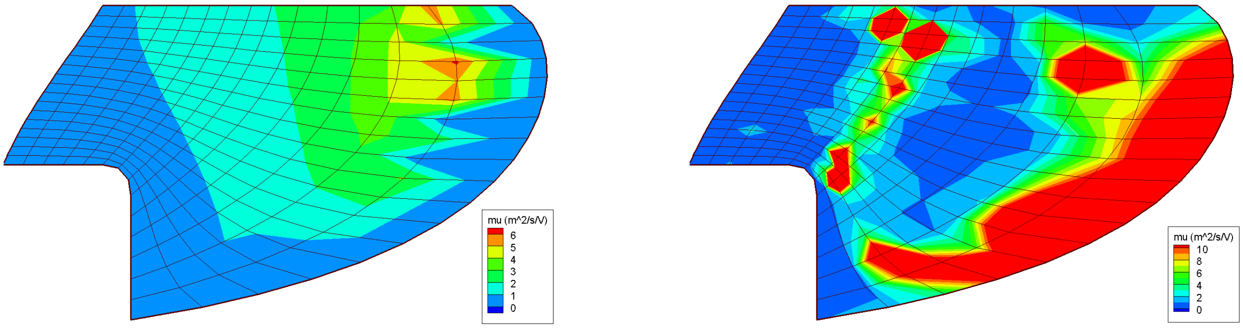
where j corresponds to a mesh cell, and the sum is performed over all particles in that cell. Unfortunately, this approach is not feasible numerically. To see why, let's consider the typical values of drift velocity v_d and the tangential velocity of electrons orbiting a field line, v_T. Using values typical of the CHT, $v_d \equiv \mu E = 5\text{m}^2/\text{Vs} \times 20,000\text{V/m} = 10^5\text{m/s}$. The tangential velocity can be approximated as $v_T = \sqrt{kTe/m} = 10^6\text{m/s}$. From this simple calculation, we can see that $v_T > v_d$. Given a finite number of particles, and the fact the particles translate along the field line during their orbit, the summation will always result in non-zero mobility. Due to the magnitude of v_T, a single unpaired term will likely wash out any actual drift velocity.

This shortcoming became obvious during our validation tests. We ran the simulation for a case in which all diffusion terms were disabled, yet the code predicted finite transport. Even more interesting was the fact that mobility was only slightly increased when collisions were included or when electric field was doubled. Hence, we implemented an alternative method of computing drift velocity. Instead of summing the axial components of particle velocity, we consider the velocity of the guiding center. We determine the guiding center as $r_g = 0.5(z^+ + z^-)$, where z^+ and z^- are the extents of particle's position during an orbit. These quantities are reset once per orbit. Drift velocity is then $v_d = r_g/\tau$, where τ is the time delta since previous sampling.

Only particles having diffused more than $1.5r_L$ were counted. Here r_L is the local Larmor radius computed using particle's tangential velocity, and the strength of the magnetic field at the particle location. Stray particles were removed from the simulation and were subsequently replaced by new particles at the field line. The radial position of the newly created particles was based on the density of the ions.

All simulations presented in this paper were executed on a Sony VAIO laptop with the Intel Core i7 2.5GHz CPU. This CPU supports up to 8 concurrent threads. Typical simulation times ranged from 400 seconds for a quick estimate with 10,000 particles and 10,000 time steps to 2600 seconds for the runs used to generate the results for this paper. These runs used 50,000 particles and ran for 20,000 time steps. Computed properties, including mobility, were then exported as a 2D mesh.

Figure 5 compares the mobility computed using our kinetic approach to that used by HPHall. Although the background values of mobility are similar quantitatively, we can see stark differences between the two versions. The kinetic solution contains two distinct regions of high mobility which are not seen in the analytical model. High production of secondary electrons was predicted by the code for the field lines at the left band. NWC may also explain the oscillatory nature of transport, which seems to be related to the high number of SEE seen on the magnetic field lines in this region. On the other hand, the high mobility in the right band may be due to a strong electric field. The right field line corresponds to the location where HPHall predicts drop in potential corresponding the the start of the acceleration zone. We also see reduced mobility near the innerpole. This region is dominated by increased magnetic pressure which reduces flux of electrons to this region. It should be noted that the results shown here are statistically accurate. We performed multiple simulations with a varying number of particles, and all simulations produced comparable results.



(a) HPHall

(b) Kinetic Code

Figure 5. Comparison of the analytical mobility used by HPHall (left) to the kinetically determined mobility from Lynx.

V. HPHall results with kinetic mobility

The mobility contour obtained in Lynx was next loaded into HPHall. We modified HPHall to load mobility values from a file. We next ran HPHall for 5,000 time steps, and exported the new 2D results. Comparison of centerline plasma properties can be seen in Figure 6. Solid markers correspond to experimental data. The dashed lines are the results obtained with the classical model, and the solid lines are the ones obtained by loading the self-consistently determined mobility. A clear improvement in plasma potential is seen. Although the potential deviates somewhat from the control point, the potential is seen to increase towards the anode, and the trough region is eliminated. Temperature results are less conclusive, partly due to the poor agreement of the initial results with the data. The peak of plasma density is seen to decrease and move towards the anode.

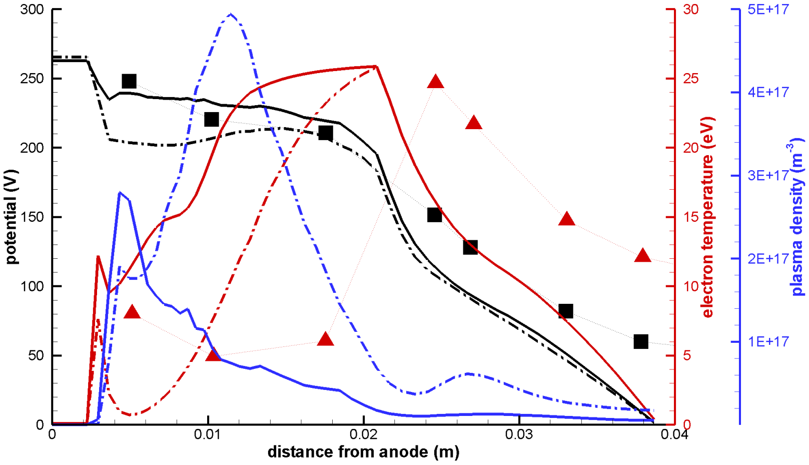


Figure 6. Plot of several plasma parameters along the thruster centerline. The markers correspond to experimental data from Ref.[15] Experimental plasma density, measured in a single point in the channel, was $\sim 6 \times 10^{17}$ m^{-3}. The results computed using the analytical mobility model are shown using the dashed lines. The results obtained with the kinetic mobility are shown with the solid lines.

A. Particle Sampling

During this simulation we also sampled particles crossing a virtual plane in the near plume region. It's difficult to describe the velocity and flux distribution function of the Hall thruster plume analytically. Unlike ion thrusters, Hall thruster discharges are open to the ambient environment, and the acceleration profile is a function of the plume itself. Sampling particles as they cross some plane is a simple and effective way to describe the velocity and flux space. By sampling velocities and the corresponding spatial position for a sufficiently large number of particles we can obtain a discretized velocity distribution function. A peculiar feature of Hall thrusters is that some acceleration occurs outside the actual thruster. In the case of the CHT, the external acceleration accounts for approximately 50eV of ion energy. In order to capture this feature, we sample the particles sufficient distance from the exit plane, in a region where potential drop becomes negligible.

VI. Plume simulations with Draco

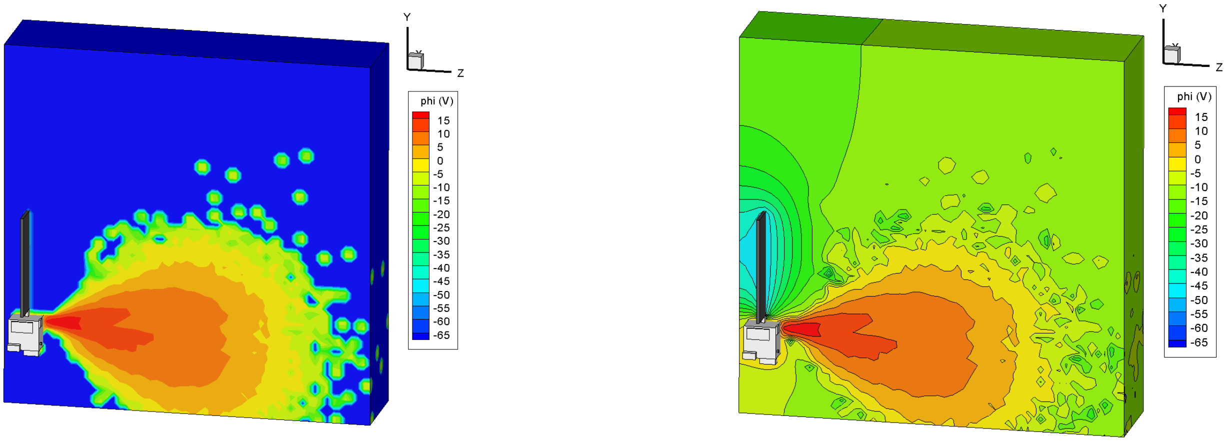
The discretized velocity distribution function is next used to model the plume environment produced by these thrusters. The plume modeling was performed using the final piece of our multiscale approach, a 3D ES-PIC code called Draco.[24] Draco was developed at Virginia Tech as a general plasma simulation tool suitable for modeling electric propulsion plumes and their interaction with spacecraft components. It readily interfaces with detailed geometries produced in COTS CAD/CAE packages. The code operates on a rectilinear mesh containing a subset of cut cells. The cut cells are generated automatically based on the user specified geometry surface mesh.

A. Source Model

Particles are injected into the simulation from source element groups. Draco supports variety of source models, including one developed specifically for interfacing with HPHall.[25] This model takes as its input a file containing a large number of $[r, v_z, v_r, v_\theta]$ tuples. The source randomly selects a tuple from the list and rotates it through a random azimuthal angle θ. Position and and velocity is also rotated according to the normal vector of the source elements. This discretized model offers the benefit of being able to capture not just a general non-Maxwellian velocity distribution space, but also radial variation in mass flux.

B. Hybrid Potential Solver

Generally two methods exist for obtaining plasma potential for plume simulations. The potential can be obtained by solving the Poisson's equation. This approach requires cell spacing small enough to capture gradients in the solution. Since the plume is quasineutral, and not magnetized, a simpler method is available based on the direct inversion of the Boltzmann relationship for electrons, $\phi = \phi_0 + kTe\ln(n_i/n_0)$. Although this approach computes the correct potential in the plume region, it does not take into account the non-neutral sheath region. In the case of a GEO satellite, the sheath is not negligible and can expand distances the scale of the spacecraft.



(a) Boltzmann Inversion

(b) Poisson Solver with QN switch

Figure 7. Comparison of electric potential solution for an identical charge density computed using the two approaches. The solution is identical in the plume region, but the QN-switched solver also resolves the sheath around the negatively charged solar panels.

Hence, we implemented a "QN switch" Poisson solver based on the previous work of Santi and Cheng.[26]

Prior to commencing the solver iterations, the solver calculates the local Debye length at each node of the simulation domain. If $4/3\pi\lambda_D^3 < V$, where V is the cell volume, the node is flagged as quasineutral and potential on it is fixed to the value obtained by the direct inversion. The Poisson solver then backfills the remaining region. Comparison between the two solutions is shown in Figure 7. The plots were generated by injecting particles for a small number of time steps without using any field solver. The simulation was then restarted for zero number of time steps, which resulted in the potential update, but no particle push. Hence, the charge densities are identical. The figure on the left shows the solution obtained by the inversion alone. The figure on the right shows the solution from the Poisson solver with QN-switch. It should be noted that a non-switched Poisson solver was not able to converge for this particular problem. As we can see, the solutions are identical in the plume region, as expected. The QN-switch approach however correctly captures the potential drop outside the negatively charged solar panel. The direct inversion method shown on left effectively compresses the sheath to the thickness given by a simulation cell. The ions are not aware of the solar panel until they reach the cell adjacent to it. This difference has a profound implication on the trajectories of the charge exchange ions, trajectories of which are primarily influenced by the electric fields between the plume and the spacecraft components.

C. Plume Results

We used this approach to model the plume environment around a generic spacecraft operating a cluster of two CHT thrusters. The results can be seen in right section of Figure 1. Prominent feature of the spacecraft is a large solar array, which in our model was assumed to float negative in respect to the bus. For simplicity, uniform potential was applied across the solar wing. The potential on the solar array was -20V, and the potential on the bus was 0V. The potential at the thruster exit was 10V. The simulation was performed on a stretched $40 \times 40 \times 46$ mesh. The mesh extended from 8 cm behind the thruster exit plane to 60 cm in front of the thruster. Reference values for the potential solver corresponded to the potential, density, and temperature at the thruster exit. The simulation took approximately 2 hours to complete. An actual detailed analysis of the plume environment produced by this thruster is reserved for future work.

VII. Conclusion and Future Work

In this paper we presented a new model for performing multiscale modeling of Hall thrusters. The main feature of our model is its ability to self-consistently determine electron mobility, plasma properties in the thruster channel, and also the plume environment induced by the thruster without relying on supercomputer resources. We demonstrated the approach on the 2.6cm Princeton Cylindrical Hall thruster. Our approach relies on a kinetic code that computes the spatial variation of mobility by considering the cyclotron motion of electrons, a 2D axi-symmetric code for thruster discharge, and a 3D plume code to model the plasma environment and contamination effects.

In this paper we used HPHall to model the thruster discharge. As part of our future work, we plan to perform the thruster modeling using our in-house designed 2D code. Primary motivation for this effort is to simplify the iterative processing needed to obtain a truly self-consistent solution. HPHall is written in C, following non object-oriented procedural implementation. HPHall also contains physics additional to what is needed to capture the thruster plasma environment. The goal for our work is to create a light-weight replacement to HPHall that utilizes modern software engineering paradigms and easily ties in with our kinetic code.

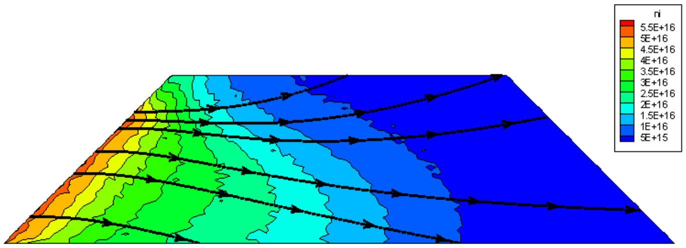


Figure 8. Plasma density and ion velocity streamlines in the sheath with converging magnetic field lines. The thruster wall forms the upper boundary, and the bottom boundary extends into the quasineutral bulk plasma.

In addition, we are investigating few additional components of Hall thruster discharge. Modern Hall thrusters, including the CHT, use magnetic fields that intersect the camber walls at off-normal angles. Configurations with highly oblique angles have been proposed to create the so-called magnetic lens, which effectively focuses ions and pushes them away from walls. However, in such a configuration, the radial component of electric field normal to the magnetic field lines can exceed the component due to the sheath drop. In that case, the net electric field points away from the walls, and the sheath collapses. We have developed a code specifically to study this near wall sheath region in highly oblique magnetic fields. A typical solution is shown in 8. The goal of this work is to produce an algebraic model that can be coupled into the thruster code to correctly compute the sheath potential drop.

VIII. Acknowledgments

This research was supported by AFOSR (Dr. Mitat Birkan is technical monitor) and NASA DC Space Grant Consortium.

References

[1] Fife, J. and Martinez-Sanchez, M., "Two-Dimensional Hybrid Particle-In-Cell (PIC) Modeling of Hall Thrusters," *24th International Electric Propulsion Conference, Moscow, Russia*, 1995, pp. 1213–1224.

[2] Parra, F., Ahedo, E., Fife, J., and Martinez-Sanchez, M., "A two-dimensional hybrid model of the Hall thruster discharge," *Journal of applied physics*, Vol. 100, 2006, pp. 023304.

[3] Hofer, R., Mikellides, I., Katz, I., and Goebel, D., "Wall sheath and electron mobility modeling in hybrid-PIC Hall thruster simulations," *AIAA Paper*, Vol. 5267, 2007.

[4] Koo, J. and Boyd, I., "Computational model of a Hall thruster," *Computer physics communications*, Vol. 164, No. 1-3, 2004, pp. 442–447.

[5] Hagelaar, G., Bareilles, J., Garrigues, L., and Boeuf, J., "Two-dimensional model of a stationary plasma thruster," *Journal of applied physics*, Vol. 91, 2002, pp. 5592.

[6] Scharfe, M., Thomas, C., Scharfe, D., Gascon, N., Cappelli, M., and Fernandez, E., "Shear-based model for electron transport in hybrid Hall thruster simulations," *Plasma Science, IEEE Transactions on*, Vol. 36, No. 5, 2008, pp. 2058–2068.

[7] Garrigues, L., Hagelaar, G. J. M., Boeuf, J. P., Raitses, Y., Smirnov, A., and Fisch, N. J., "Two Dimensional Hybrid Model of a Miniaturized Cylindrical Hall Thruster," *30^{th} International Electric Propulsion Conference*, Florence, Italy, 2007, IEPC-2007-157.

[8] Sydorenko, D., Smolyakov, A., Kaganovich, I., and Raitses, Y., "Modification of Electron Velocity Distribution in Bounded Plasmas by Secondary Electron Emission," *IEEE Transactions on Plasma Science*, Vol. 34, No. 3, 2006, pp. 815–824.

[9] Hirakawa, M. and Arakawa, Y., "Particle simulation of plasma phenomena in Hall thrusters," *Proceedings of the 24th International Electric Propulsion Conference (Moscow), IEPC-95-164*, 1995.

[10] Szabo, J., Martinez-Sanchez, M., and Batishchev, O., "Fully kinetic Hall thruster modeling," *IEPC Paper*, 2001, pp. 01–341.

[11] Fox, J., *Advances in Fully-Kinetic PIC Simulations of a Near-Vacuum Hall Thruster and Other Plasma Systems*, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[12] Sullivan, K., *PIC simulation of SPT Hall thrusters: high power operation and wall effects*, Ph.D. thesis, Massachusetts Institute of Technology, 2004.

[13] Raitses, Y. and Fisch, N., "Parametric investigations of a nonconventional Hall thruster," *Physics of Plasmas*, Vol. 8, 2001, pp. 2579.

[14] Brieda, L., Keidar, M., Raitses, Y., and Fisch, N., "Self-Consistent Calculation of Electron Transport in a Cylindrical Hall Thruster," *31st Internation Electric Propulsion Conference, Ann Arbor, MI*, 2009, pp. 1–9.

[15] Raitses, Y., Smirnov, A., and Fisch, N., "Effects of enhanced cathode electron emission on Hall thruster operation," *Physics of Plasmas*, Vol. 16, 2009, pp. 057106.

[16] Smirnov, A., Raitses, Y., and Fisch, N. J., "Plasma measurements in a 100W cylindrical Hall thruster," *Journal of Applied Physics*, Vol. 95, No. 5, 2004, pp. 2283–2291.

[17] Brieda, L., Keidar, M., Raitses, Y., and Fisch, N., "Investigation of Electron Transport in a Cylindrical Hall Thruster using a Kinetic Code," *45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Denver, Colorado, Aug. 2-5, 2009*, 2009, pp. 1–8.

[18] Wikipedia, "JAVA Performance," `http://en.wikipedia.org/wiki/Java_performance#Comparison_to_other_languages`.

[19] Particle In Cell, "Get Results Faster with Java Multithreading," `http://www.particleincell.com/2011/java-multithreading`.

[20] Lieberman, M. and Lichtenberg, A., *Principles of plasma discharges and materials processing*, Wiley-interscience, 2005.

[21] Nicklass, A., Dolg, M., Stoll, H., and Preuss, H., "Ab initio energy-adjusted pseudopotentials for the noble gases Ne through Xe: Calculation of atomic dipole and quadrupole polarizabilities," *Journal of Chemical Physics*, 1995.

[22] Szabo Jr, J., *Fully kinetic numerical modeling of a plasma thruster*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[23] Sydorenko, D., *Particle-in-cell simulations of electron dynamics in low pressure discharges with magnetic fields*, Ph.D. thesis, University of Saskatchewan, 2006.

[24] Brieda, L., Kafafy, R., Pierru, J., and Wang, J., "Development of the draco code for modeling electric propulsion plume interactions," *Proceedings of the 40th Joint Propulsion Conference, Fort Laudardale, FL, USA*, 2004, pp. 1–21.

[25] Nakles, M., Hargus, W., and VanGilder, D., "Comparison of Numerical and Experimental Near-Field Ion Velocity Distributions of the BHT-200-X3 Hall Thruster," *42nd Joint Propulsion Conference, Sacramento, CA*, 2006, AIAA-2006-4479.

[26] Santi, M., Cheng, S., Celik, M., M., M.-S., and J, P., "Further Development and Preliminary Results of the AQUILA Hall Thruster Plume Model," *39th Joint Propulsion Conference, Huntsville, Alabama*, 2003.