Numerical Model for Molecular and Particulate Contamination Transport

Lubos Brieda*
Particle In Cell Consulting LLC, Westlake Village, CA 91362

A new model for simulating molecular and particulate contamination transport is described. This model is implemented in the Contamination Transport Simulation Program (CTSP). The code utilizes techniques from rarefied gas and plasma modeling to concurrently trace multiple simulation particles representing molecular or particulate contaminants. This kinetic approach allows the code to consider external forces and inter-particle interactions. Macroscopic properties such as contaminant plume partial pressure can also be computed. CTSP implements various contamination-specific material sources including detailed models for molecular outgassing and particulate detachment. The models are described in detail. The model is demonstrated with four examples focusing on mass transport between two parallel plates, computation of the steady-state equilibrium in a closed vessel, characterization of a test article outgassing rate from a QCM measurement, and the use of purge gas to reduce the particulate fallout in atmospheric conditions. Simulation results are compared to analytical models.

Nomenclature

A	=	Area (m^2)
A_d	=	Cross-sectional area for drag force (m^2)
C	=	Particulate loading slope (dimensionless)
C_d	=	Drag coefficient (dimensionless)
C_0	=	Power law model coefficient ($kg/m^2/s^{0.5}$)
D	=	Diffusion coefficient (m^2/s)
E_a	=	Process activation energy (kCal/mol)
F	=	Force (N)
L	=	Particulate loading level (dimensionless)
M	=	Total mass contained within some region (kg)
N	=	Number of particles or samples

*President, AIAA Senior Member,

R	=	Gas constant ($\approx 1.987 \times 10^{-3}$ kCal/K/mol) or a random number in [0,1)
T	=	Temperature (K)
w_{sp}	=	Specific weight, the number of real particles represented by each simulation particle
a_d	=	Detachment acceleration (m/s^2)
d	=	Particle diameter (m or μm)
k	=	Boltzmann constant ($\approx 1.381 \times 10^{-23}$ J/K)
l	=	Particle length (m or μm)
m	=	Particle mass (kg)
n	=	Number density (1/m^3)
r	=	Molecular radius (m)
v	=	Velocity (m/s)
x	=	Position (m)
α_{sc}	=	Sticking coefficient [0,1]
Δt	=	Simulation time step (s)
Γ	=	Number flux (kg/m^2/s)
γ	=	Partition coefficient (1/m)
Φ	=	Release probability [0,1]
ρ	=	Mass density (kg/m^3)
τ_r	=	Molecular surface residence time (s)
τ_0	=	Vibrational period (s)
θ	=	Surface mass density (kg/m^2)
molecule	=	Collection of one to hundreds of atoms with sizes of order 10^{-10} m
particulate	=	Collection of billions of atoms with sizes larger than 10^{-6} m
particle	=	Arbitrary point object with some position, velocity, size, and other properties

I. Introduction

Spacecraft instruments and thermal control devices are generally highly sensitive to contamination. Contamination is any foreign material that manifests as molecular film or as microscopic particulates. Molecular contaminants arise primarily from unspent reactants used in the production of organic materials for circuit boards, harnessing, or lubricants. When exposed to vacuum, these molecules diffuse out of the native material and outgas into the local environment where they may deposit on sensitive components. Molecular films just a few hundred monolayers thick can lead to a significant transmission loss in optical instruments. This effect is particularly pronounced in the ultraviolet

spectrum, since typical molecular contaminants have a higher absorptance at the UV wavelengths[1, 2]. Performance of thermal control devices is also degraded as contaminants generally darken under UV radiation[3, 4]. On the other hand, particulates arise from the fallout of general atmospheric "dust" pollutants, from skin and clothing shed by laboratory technicians, and from flaking of thermal coatings. Particulate sizes generally range from 10 to 1,000 μm, although larger, high aspect-ratio fibers may also be present. They are present on all surfaces. Vibrational events encountered during launch or deployments may lead to their redistribution. Particulates block or scatter light, and contribute to glare in optical images. They damage mechanical actuators and thin foils, and can lead to shorts in high voltage instruments[5]. Despite best efforts taken during fabrication, integration, and testing, it is simply not possible to completely eliminate all sources of contaminants. Transport modeling then becomes an essential tool in the arsenal of the systems engineer. It can be used to predict the end of life deposition levels given some source rates. Conversely, given the allowable end of life loading, contamination modeling can be used to derive cleanliness requirements to be met prior to launch.

Historically, different methodologies have been applied to molecular and particulate contaminants. Molecules, due to their tiny size and mass, are assumed to be unaffected by gravity. They are also assumed to remain neutral. Vacuum chamber testing generally occurs in the free molecular flow regime in which the Knudsen number $Kn = \lambda/L \gg 1$. This ratio compares the distance traveled by molecules between collisions λ to some characteristic length L, such as the diameter of the vacuum chamber. In the free molecular flow, the distance between collisions greatly exceeds the characteristic length and molecules are more likely to strike the chamber walls than each other. Inter-molecular collisions can then be ignored, and molecules are assumed to travel in straight line trajectories, just like photons. This formulation leads to the widespread use of radiation heat transfer codes for contamination modeling[6]. The mass transport analyses reduces to the task of computing gray body view factors between sensitive surfaces and the outgassing sources.

The radiation-based approach suffers from a number of limitations. By imposing straight line trajectories, it becomes impossible to consider the influence of forces on the contaminants. Molecules moving away from the spacecraft may become ionized by solar radiation and subsequently back flow to the spacecraft[7, 8]. This effect is especially important in the GEO environment where the plasma sheath can extend into hundreds of meters[9] and spacecraft potentials can reach thousands of volts [10]. Furthermore, thermal re-emission is computed assuming uniform surface element temperature, which does not accurately represent regions with spatial non-uniformities in contaminant mass loading. The free molecular flow assumption also does not hold at higher pressures such as during chamber repressurization. Gravitational and aerodynamic forces are non-negligible for the much larger particulates, and therefore their transport cannot be modeled using the radiation-based approach. Historically this required development of specialized tools for each contributing factor. Some examples include the view factor based MTM code for molecular contamination[11], the ESR code[12] for estimating electrostatic return to satellites modeled as charged spheres, the particle tracing code Mastram[13] for modeling particulate redistribution, and the use of Direct Simulation Monte Carlo (DSMC) codes for the transition regime[14]. In order to address these shortcomings and the need for multiple tools, a unified

Contamination Transport Simulation Program (CTSP) has been developed. To our knowledge, CTSP is the first and the only code capable of self-consistently simulating both molecular and particulate mass transport in complex systems while supporting time-variant external forces and environmental parameters. The code is described in this paper. The common core is introduced first. The models used for contamination generation, transport, and surface impingement are discussed next. The code is then demonstrated with four examples. The first example compares simulated view factors between two square plates to an analytical model and compares deposition results to a heritage approach. The second example considers outgassing in a closed cavity and demonstrates that concentration gradients vanish at steady state. The third example models a common experimental procedure of using a Quartz Crystal Microbalance (QCM)[15] to determine the outgassing rate of a test article placed in a vacuum chamber. Finally, the use of purge gas to reduce the fallout of particulates onto a detector is considered.

II. Code Overview

CTSP is a kinetic code in which the molecular and particulate contaminants are represented by simulation particles. A simulation particle is a general numerical construct for a point object having some known position, velocity, and other parameters of interest, such as mass, charge, or cross-sectional area. The code does not distinguish between molecular and particulate particles, with the exception of the surface impact behavior. The physical domain is described using a triangular and/or quadrilateral surface mesh. This mesh is generated in external off-the-shelf CAD/FEM tools. CTSP supports various common mesh formats, including Universal and Nastran, as well as Thermal Synthesizer (TSS) assemblies used by the thermal community, and Stereolithography (STL) files. The STL format describes the geometry using triangles, with the distinction that these files are exported directly from CAD packages. The surface tessellation is not optimized for numerical analysis, but it allows the user to simulate highly complex geometries without having to mesh them first. The code has been applied in this fashion to models with over 2 million surface elements[16].

Contaminants are injected continuously into the computational domain from user-specified surface regions according to mass generation models described in the following section. Computationally, it is not possible to directly simulate every single molecule or a particulate present in a system with real-world dimensions. Even at the low 10^{-6} Torr pressure found in high end vacuum chambers, there are over 3×10^{16} molecules per cubic meter. Just the memory required to store their positions and velocities is orders of magnitude beyond current supercomputer capabilities. Therefore, a stochastic approach, similar to one found in Direct Simulation Monte Carlo (DSMC)[17] and Particle in Cell (PIC)[18] codes used by the rarefied gas and plasma community, is used. Each simulation particle corresponds to w_{sp} real molecules or particulates with an identical position, velocity, mass, and other properties. The term w_{sp} is known as specific weight. The initial properties of each simulation particle are sampled from the real distribution. For instance, the molecular injection velocity is selected randomly from the Maxwellian distribution function at the surface temperature. The number of simulation particles to inject at each time step is computed from the desired mass flux or the particulate

detachment rate. The number of simulation particles scales inversely with w_{sp}. At lower weights, the simulation contains more particles (and hence more stochastic samples), resulting in a reduced numerical noise at the expense of an increased run time. The number of simulation particles has no quantitative impact on results, but it influences their quality.

Particle positions and velocities are integrated through small Δt time steps. First, the velocity $\vec{v}$ of each simulation particle is updated from $d\vec{v}/dt = \vec{F}/m$. The term on the right hand side is the sum of all forces acting on the particle, and m is the particle mass. The fourth-order Runge-Kutta method is used to perform this integration. In general, molecular transport analyses are performed with $\vec{F} = 0$ in which case, the velocity remains constant. The particle position $\vec{x}$ is then advanced from $d\vec{x}/dt = \vec{v}$ using the Forward Euler method. At each time step, it is necessary to check for particle-surface interactions. During the push, each particle moves in a straight line to a new position offset by $\vec{v}\Delta t$ from its current position. Numerically, testing for particle impacts reduces to performing line-triangle or line-quadrangle intersection checks. The brute-force approach in which each particle is checked against the entire surface mesh would be highly computationally taxing given that a typical simulation may contain millions of surface elements and hundreds of thousands of computational particles. Instead, the surface mesh is stored internally in an octree, allowing the code to efficiently retrieve the subset of surface elements in the vicinity of the particle. Multithreading is also implemented to perform these checks in parallel. If an intersection is found, the particle is first pushed to the surface. The new post-impact velocity is then computed, assuming the particle does not "stick" to the surface. The particle position is then advanced through the reminder of the time step. This algorithm can thus handle multiple surface impacts during a single time step. There are several factors to consider when selecting the value of Δt. The motion of molecules in the free-molecular flow regime is not affected by the time step size - this is in fact the motivation behind ray tracing algorithm. However, usually we desire sufficiently low values for visualization purposes to generate smooth animation plots. Given that typical molecular thermal speeds are around 300 m/s, $\Delta t = 10^{-4}$ s will result in molecules moving approximately 3 cm per step. Since CTSP checks intersections with all surfaces within the box bounded by the particle starting and ending position, the total computational time may also actually be decreased using a higher number of smaller time steps. The presence of external forces, or the numerical integration of mass outgassing may introduce additional constraints on the time step size.

CTSP supports gravitational, aerodynamic, electrostatic, and solar radiation pressure forces. As the code does not contain built-in field solvers, it is necessary to import the vector or scalar data needed by these terms. The code supports two mechanisms for doing so. First, spatially invariant "world" properties can be set. These properties can be constant or time varying. Secondly, it is possible to import one or more comma-separated files containing solutions from external solvers, such as velocity vectors and pressures from a CFD solver, or the electric field from a charging computation. This field data is interpolated onto the particle position during the velocity integration step. PIC and DSMC codes utilize a volume mesh to perform a similar interpolation. CTSP, on the other hand, does not use a volume mesh for the

particle push. Volume mesh, if specified, is used solely to compute macroscopic gas properties such as pressure or the mean velocity for post-processing. In free molecular flows, there is no physical quantity limiting the maximum cell size. The user could thus set the volume mesh arbitrarily, resulting in an artificial dependence of the solution on the mesh quality. For instance, interpolating an imported CFD solution onto an overly coarse mesh would result in the loss of fine details in regions with rapidly changing properties, such as in the near-surface boundary layer. On the other hand, retaining the imported mesh is also impractical as unstructured meshes are often used in CFD. Such meshes are not well-suited to kinetic computations as it is numerically expensive to trace particles on unstructured meshes. CTSP thus retains only the vertexes of the imported solution and stores the associated data in another octree. Force calculation is performed by interpolating data from this point cloud of field quantities using the inverse distance weighting method, $u(\vec{x}) = \sum_{i=1}^{N} w_i(\vec{x})u_i / \sum_{i=1}^{N} w_i(\vec{x})$ with $w_i(\vec{x}) = d(\vec{x}, \vec{x}_i)^{-p}$. Here $(\vec{x}_i)u_i$ are the positions and values of the data points in the vicinity of the particle located at $\vec{x}$, d is the distance between each point and the particle, and p is an interpolation smoothing factor. The user can specify multiple files corresponding to different time intervals with linear interpolation used for the intermediary intervals.

III. Mass Generation

CTSP implements various material sources applicable to contamination modeling, including detailed models for molecular outgassing, effusion[19], liquid droplet evaporation[20], plume expansions[21], particulate redistribution, and a random coverage by fibers. Of these, the molecular outgassing and particulate sources are the most commonly used and are described next.

A. Molecular Outgassing

Molecular outgassing arises from volatile materials trapped inside the bulk substrate diffusing to the surface and desorbing into the gas phase. The time evolution of the trapped population is given by the diffusion equation,

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D \nabla \rho) \tag{1}$$

where D is the diffusion coefficient in m^2/s, and ρ is the contaminant mass density in kg/m^3. The diffusion equation is however not practical to contamination studies due to the presence of the spatial derivative on the right hand side. In order to numerically integrate it, we would require a discretized mesh for the internal structure of all outgassing hardware articles. Therefore, a well-mixed approximation is commonly used, in which we assume that the contaminant concentration within the material can be described by some uniform value[22]. Then, integrating through a control volume, we obtain

$$\frac{dM}{dt} = -\Gamma A \tag{2}$$

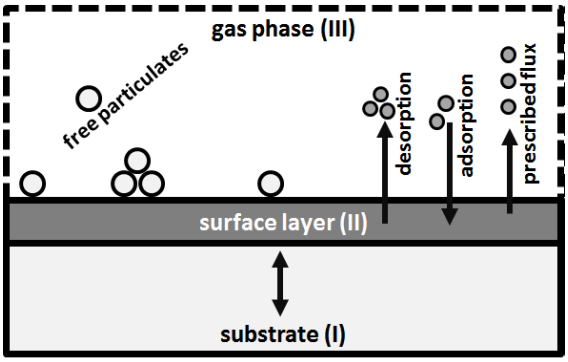


Fig. 1 Overview of the surface model used by CTSP. All components are assumed to consist of a solid substrate containing some trapped contaminants and a surface layer in contact with the gas phase.

where $M = \int_V \rho dV$ (kg) is the total contaminant mass in the control volume, and $\Gamma A = -\oint_S D(\partial\rho/\partial\hat{n})dA$ is the outgassing mass flow rate given in kg/s. A is the exposed surface area, and Γ is the mass flux in kg/m^2/s. In the case of a homogeneous diffusion coefficient and uniform concentration gradient, we can write $\Gamma = -D(\partial\rho/\partial\hat{n})$. As can be seen from this relationship, the outgassing rate varies with time. As the amount of trapped material becomes depleted, the concentration gradient also decreases, resulting in a reduction of the outgoing mass flux. The rate also scales with temperature. In general, the diffusion coefficient follows the Arrhenius equation, $D = D_0 \exp(-E_a/RT)$, where E_a is the process activation energy (in kCal/mol), R is the universal gas constant (in kCal/mol/K) and T is the local temperature (in K). Because of this relationship, it is customary to perform vacuum bakeouts of components prior to the final assembly to speed up the removal of trapped contaminants.

CTSP attempts to capture this time and temperature dependent behavior. As plotted in Figure 1, all test objects are assumed to consist of a native bulk substrate (Region I) and a thin surface layer (Region II). Both regions contain an arbitrary heterogeneous combination of molecular species. The surface layer may also contain particulates. Molecules and particulates leaving the surface layer enter the gas phase (Region III). They eventually encounter other geometry components (unless they exit the simulation through open boundaries) and possibly deposit onto that component surface layer (Region IV / foreign Region II). Molecules can diffuse from the surface layer into the substrate (Region V / foregin Region I) if the surface concentration exceeds the bulk concentration. For generality, we also allow the surface to generate an additional prescribed flux. This term could represent the degradation of the surface by atomic oxygen or some chemical process, but it is also useful when QCM-measured outgassing rates are available. The initial composition of the substrate and surface regions for each geometry component is specified in the simulation input file.

Equation 2 can be integrated in time if an expression for the mass flux Γ is known. CTSP implements two models for this term. First, the code supports a simple power law model commonly used by the vacuum community[2]

$$\Gamma = C_0 \exp\left(\frac{-(E_a)_{dif}}{RT}\right) t^k \tag{3}$$

where $(E_a)_{dif}$ is the activation energy for diffusion and $k = -0.5$ for a diffusion-limited process. In this formulation, C_0 is a scaling coefficient used to correlate the model to experimental data obtained from an outgassing characterization test such as the ASTM-E1559[23]. This power law model does not support mass diffusion from the surface layer back to the solid, nor does it take into account the existing surface concentration.It is derived by solving the diffusion equation in a semi-infinite one-dimensional medium initially at mass density ρ_0 and having a surface at $x = 0$ maintained at ρ_1. Crank[24] shows that the solution for such a system is given by

$$\frac{\rho - \rho_1}{\rho_0 - \rho_1} = \text{erf}\frac{x}{2\sqrt{Dt}} \tag{4}$$

The surface flux is obtained by differentiating the above equation in respect to x and evaluating at $x = 0$. Utilizing the definition of the error function, $\text{erf}(x) = 2/\sqrt{\pi}\int_0^x \exp(-t^2)dt$, and assuming that the surface concentration $\rho_1 = 0$, we obtain

$$\left(D\frac{\partial \rho}{\partial x}\right)_{x=0} = \frac{D\rho_0}{\sqrt{\pi Dt}} = At^{-0.5} \tag{5}$$

CTSP also implements a detailed model for the well-mixed approximation. The flux term in Equation 2 quantifies the amount of mass lost by the solid region. To satisfy mass conservation, it also governs the number of molecules gained by the surface layer. The surface layer is also incremented by adsorption of molecules from the gas phase and is similarly depleted by desorption of the surface film into the gas phase. We label these two terms Γ_a and Γ_d. The time evolution of the surface mass density θ (kg/m^2) is thus given by

$$\frac{d\theta}{dt} = -D(\partial\rho/\partial\hat{n}) + \Gamma_a - \Gamma_d \tag{6}$$

Next, following the approach taken by Fang, et.al.[22], we assume that the diffusion flux $-D(\partial\rho/\partial\hat{n})$ is proportional to the amount of material inside and on the surface of the object. We define a sorption function $H(\rho_c, \theta)$ such that

$$-D\frac{\partial \rho}{\partial \hat{n}} \equiv H(\rho, \theta) = h\,(\rho - \gamma\theta) \tag{7}$$

Here $h = Dk_1$ with k_1 (m^{-1}) being a "diffusion length coefficient". The term γ (m^{-1}) is an "equilibrium partition coefficient" such that at equilibrium $\rho = \gamma\theta$. Both parameters are assigned the value of 1 in this paper. We thus have $d\theta/dt = h(\rho_c - \gamma\theta) + \Gamma_a - \Gamma_d$. From this expression we see that at equilibrium we also require $\Gamma_a = \Gamma_d$. Satisfying this requirement is demonstrated in Example 2. The desorption flux is given by $\Gamma_d = \theta_{ml}/\tau_r$ where

$$\tau_r = \tau_0 \exp\left(\frac{(E_a)_{des}}{RT}\right) \tag{8}$$

is the molecular residence time. The parameter τ_0 is the vibrational period of the molecule with typical values around 10^{-13} s [2] and $(E_a)_{des}$ is the activation energy for the desorption process. The parameter $\theta_{ml} = \min(\theta, \eta_0/(\pi r^2))$ limits the maximum desorption rate to one given by a fully occupied monolayer. In this expression, $\eta_0/(\pi r^2)$ is the maximum number of molecules per unit area, r is the molecular radius and η_0 is a scaling "packing" factor. The adsorption flux Γ_a term is described in Section IV.

The above algorithm is implemented numerically as follows. We start by looping through all surface components. On each, we first use the power law or the detailed model to compute the total number of molecules diffusing to (or from) the surface layer, N_1. The mass lost or gained by these molecules is given by $\Delta M = mN_1$. We then loop through all surface elements, and transfer a fractional number of molecules to that element's surface layer. The fraction is given by the ratio of the element area to the total component area obtained from the surface tessellation. Instead of storing θ, CTSP tracks the actual mass of molecules on each surface element $(M_2)_i = \theta_i A_1$. Next we compute the number of real molecules to desorb, $N_d = (M_2/(m\tau_r))\Delta t$. The corresponding number of simulation particles is $N_{sim} = N_d/w_{sp}$. Generally, N_d will not be evenly divisible by the specific weight. The code supports two injection schemes. In the exact scheme, as many particles as possible will be created with the default weight w_{sp} and then an additional particle will be created with some fractional weight. The second stochastic approach does not create fractional weight particles but instead uses a random number to create full weight particles with probability N_d/w_{sp}. This second approach will be mass conserving only on average at the steady state but avoids the excessive number of simulation particles that may result with the first exact model.

B. Particulates

Just as with molecular contaminants, simulating particulate redistribution requires models for generation, transport, and deposition. Particulate contamination is traditionally divided into two categories: "standard" particulates and fibers. Fibers are large particulates with length exceeding 1,000 μm and having aspect ratios $AR \equiv l/d > 10$[25]. Fibers can be characterized by specifying their count per unit area along with the observed size ranges. CTSP implements a source for fibers which generates the user specified surface concentration with lengths and aspect ratios sampled from the uniform distribution at user specified limits. The model for the standard particulates is more involved, and is described next.

Particulate contaminants vary greatly in size and shape. Generally, they are non-spherical and have an aspect ratio increasing with their size. Directly characterizing the size and shape of each particulate is not feasible since even a well-cleaned surface contains tens of millions of particulates per unit area. The contamination control community therefore uses the IEST-STD-1246D standard to describe the size variation of surface particulates[26]. This standard provides a cumulative distribution function given by

$$\log_{10}(N_{cum,0.1}) = C\left[\log_{10}^2(L) - \log_{10}^2(l)\right] \quad l \geq 1 \tag{9}$$

Here $N_{cum,0.1}$ is the total number of particulates with sizes $\geq l$ per 0.1 m^2 (prior versions of this standard used the same model but the count was per ft^2). The particulate length l is given in μm. The parameter L is the surface cleanliness level and C is the "slope" of the distribution. Visibly clean surfaces have $L \approx 500$. Since for $l > L$ the term in the parentheses becomes negative, the level also corresponds to the largest particle size encountered with the frequency of one particle per 0.1 m^2. The standard assumes $C = 0.926$ for freshly cleaned surfaces, however real-world tape lifts indicate values closer to 0.4[2]. Both C and L are user inputs. We are also generally more interested in the actual number of particulates of a given size per m^2. This value can be approximated by subtracting two cumulative counts offset by 1 μm and multiplying the result by 10,

$$N = 10 \cdot 10^{C\left(\log_{10}^2(L) - \log_{10}^2(l)\right)} - 10^{C\left(\log_{10}^2(L) - \log_{10}^2(l+1)\right)} \tag{10}$$

The amount of particulate contamination can alternatively be quantified using the percent area coverage (PAC). This value specifies the fraction of the surface obscured by particulates and can be measured directly by optical instruments. Clearly, the two descriptions of surface loading need to be consistent with each other, implying that

$$PAC = \frac{\sum_{l=1}^{1000} N_l A_l}{1\text{m}^2} 100\% \tag{11}$$

The upper limit on the sum arises from larger particles generally characterized as fibers and counted separately. The term A_l is the surface obscuration area of the typical particulate with length l. Raab[27] and Ma, Fong, and Lee[28], performed comparisons between particulate counts and the PAC, and found the best agreement was reached with shapes approximated as cylinders with spherical caps. The aspect ratio ($AR = l/d$) varies with particle size. As summarized by Perry[29],

$$AR = \begin{cases} l^{0.1088} & l \in [1, 69] \\ l^{0.8804}/26.53 & l \in [70, 175] \\ l^{2.589}/181500 & l \in [176, 346] \\ l^{0.8964}/9.138 & l \geq 347 \end{cases} \tag{12}$$

The cross-sectional area is then

$$A_l = d(l - d) + \pi d^2/4 \tag{13}$$

The corresponding volume is $V = (l - d)\pi d^2/4 + \pi d^3/6$. This volume is used to set the particulate mass from the material density, $m = \rho V$. Note that here A_l corresponds to the cross-sectional area parallel to the particle long axis, corresponding to the particle lying flat on the surface. For aerodynamic drag computations, we assume that the particle aligns with the flow so that $A_d = (\pi d^2)/4$.

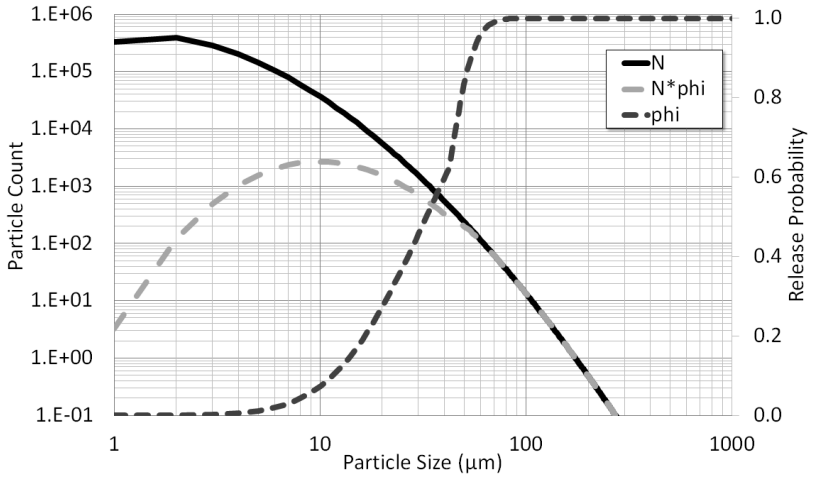


Fig. 2 Surface particle counts, release probability, and the number of detached particles for $L = 400$, $= 0.926$ and $a = 5$ g.

The distribution given by Equation 10 indicates that the concentration of particulates increases exponentially as the particle size decreases. However, small particulates are also less likely to detach due to an increased ratio of adhesion to detachment forces. Particulates unable to detach do not contribute to contaminant redistribution. Determining what fraction of particulates of size l detaches remains a significant uncertainty in our model. In 1987, Klavins and Lee studied the problem of surface adhesion by applying static loads to a test sample placed in a centrifuge[30]. These measurements were performed at loads up to 10^5 g and showed a large variation in the detachment probability between individual tests. This variation is expected, since the probability that a particle detaches is strongly influenced by the local surface roughness, particle shape, size, and orientation. Environmental effects such as humidity and electrostatic charge also play a role. Hence, at best, only a simple macroscopic estimate of detachment probability Φ can be made. The authors found it to follow

$$\Phi = [1 + \text{erf}(\log(a_d/a_m)/\sqrt{2}\sigma_0)]/2 \tag{14}$$

where a_d is the applied acceleration and $\sigma = 1.45$ is the standard deviation. The parameter a_m is the mean acceleration for a 50% removal, and is given by $(85.07/L)^{4.08}$ for particles smaller than 42 μm and $(52.37/L)^{13.6}$ otherwise.* The distribution of particles released from the surface can then be obtained by multiplying the initial size distribution with the release probability, $N \cdot \Phi$. These expressions are visualized in Figure 2 for $L = 400$, $C = 0.926$, and $a = 5$ g. As can be seen, the release model predicts all particles larger than 100 μm detach given the 5 g acceleration. On the other hand, the detachment probability is less than 30% for particles smaller than 20 μm. The light gray dashed line is the distribution that needs to be generated by the particulate source.

For generality, we prefer to evaluate Φ with a_d obtained from the sum of forces acting at the injection location. We

*These expressions are listed as $(52.37/x)^{4.08}$ and $(85.07/x)^{13.6}$ in [30] due to an apparent typo. The formulation shown here is consistent with graphs in the reference paper.

could generate simulation particles by sampling sizes from Equation 10 and for each, computing the release probability. This approach is however not practical. Considering $L = 400$ and $C = 0.926$, we see that for every 200 μm particle, there are over 13,000 20-μm particles. We thus need to sample, on average, at least 13,000 20-μ m particles to obtain a single 200 μm particle. The dynamics of the large and small particles are sufficiently different making it important that all sizes be represented. Attempting to generate a sufficient number of large particulates would result in the simulation becoming saturated by the smaller constituents. Therefore, a better approach is to divide the population into several bins and generate a constant number of simulation particles per bin. The particle specific weight w_{sp} can be used to recover the original distribution function. CTSP uses bins of the following sizes: [1,10), [10,25), [25,50), [50,100), [100,250), [250, 500), [500,750), [750,1000). In each bin, particle sizes are sampled from the uniform distribution. This approach assures that the simulation contains a statistically significant number of particles off all sizes. Simulations presented in this paper used 100 particles per bin for a total of 800 particles per surface element. All particles in a single bin share the same specific weight. The weight is set such that the total percent area coverage represented by the particles in the bins equals the PAC given by Equation 10. In each $[l_1, l_2)$ bin, we first sample N_p random sizes, and compute

$$w_{sp} \sum_{p}^{N_p} A_p = A_{ele} \sum_{l=l_1}^{l_2-1} N_l \cdot A_l \tag{15}$$

Here A_p is the cross-sectional area of the p-th particle and A_{ele} is the surface area of the surface element. N_l is given by Equation 10 and A_l is the area of a particle of size l per Equation 13.

Particles generated by the above algorithm are initially attached to the surface. The code next iterates over all particles and for each, computes the detachment probability Φ from Equation 14 or from a constant value specified by the user. The particle is detached if $\Phi \geq R$, where R is a random number. Otherwise, the particle cross-sectional area is used to update the source element percent area coverage. The model of Klavins and Lee does not offer any insight into the detachment rate. It is not clear whether particles detach instantly or whether the detachment takes place over an extended period of time. The source model thus implements two schemes. In the first one, all particles able to leave do so at the first time step. The second approach models uniform detachment rate over a finite period of time. In this approach, Φ/k_d is compared to a random number, with k_d being the number of time steps over which detachment is considered. This approach is useful in simulations with time-varying gravitational or aerodynamic environments, such as those encountered in payload fairing during spacecraft launch. In analysis, we compute particulate redistribution using both approaches and retain the worst-case prediction from each scenario.

IV. Particle Motion and Surface Impact

Once particles are generated by their respective sources, their positions are updated by numerically integrating the equations of motion. The total force acting on each particle is $\sum \vec{F} = \vec{F}_g + \vec{F}_e + \vec{F}_d$ where the terms correspond to

gravitational, electrostatic, and aerodynamic drag forces. Orbital motion of the parent body and solar pressure can also be included to model particulate return on orbit crossings. The following expressions are used to evaluate these terms:

$$\vec{F}_g = m\vec{g} \tag{16}$$

$$\vec{F}_e = q\vec{E} \tag{17}$$

$$\vec{F}_d = \frac{1}{2}\rho_g C_d A_d |\vec{v}_a - \vec{v}_p|(\vec{v}_a - \vec{v}_p) \tag{18}$$

$$\tag{19}$$

In the electrostatic Lorentz force, q is the particle charge and $\vec{E}$ is the electric field. Model of White[31] is used to compute the drag coefficient,

$$C_D = \frac{24}{Re} + \frac{6}{1+\sqrt{Re}} + 0.4 \qquad Re < 2 \times 10^5 \tag{20}$$

The code defines the Reynolds number as $Re = \rho_g u l/\mu$ where ρ_g is the gas mass density, u is the magnitude of the relative velocity between the particle and the ambient gas, l is the particle major length, and μ is the dynamic viscosity of the gas. A_d is the cross-sectional area for drag computations. To improve performance, the code applies each force only if the appropriate data are specified. For instance, the gravitational force is considered only if the world acceleration vector $\vec{g}$ is nonzero. Aerodynamic drag is computed only if $\rho_g > 0$. The drag force is also only applicable to particulates in the continuum regime in which $Kn \ll 1$. The mean free path $\lambda = (\sigma n)^{-1}$ for a nitrogen molecule at atmospheric pressure and room temperature is approximately 68 nm. This value was obtained from $\sigma = \pi(2r)^2$ and $n = P/kT$ where $r = 155$ pm and k is the Boltzmann constant. The Knudsen number for a 10 μm particle (the smallest size encountered in typical simulations given the large forces needed to dislodge smaller particles) is thus $K_n = \lambda/l = 6.8 \times 10^{-3} \ll 1$. The result is even smaller for the larger particles. From the particle's frame of reference, the flow is well in the continuum regime. The above calculation was made for the atmospheric pressure, which is the typical environment in which particulate fallout is considered. The only exception is launch in which the pressure decays with altitude. We can expect that all particulates that detach will do so between launch and the "max Q" when the dynamic pressure on the spacecraft is maximized. Depending on the launch vehicle, "max Q" occurs at altitudes between 11 and 14 km. The atmospheric pressure at 14 km is approximately 2.7× smaller than at the sea level. This corresponding 2.7× increase in the Knudsen number still places the flow well in the continuum regime, $(K_n)_{maxQ} = 1.87 \times 10^{-2} \ll 1$ for the 10 μm particulate. In the case of the molecular contaminants, intermolecular interactions outside the free molecular flow regime can be modeled using the DSMC algorithm[17]. The CTSP DSMC implementation is described in more detail in [32].

During each particle push, the code checks for surface interactions. If the particle strikes a surface, an impact handler determines whether the particle sticks to the surface, and if not, sets its post-impact velocity. The surface impact is the main part of the code where a different algorithm is applied to particulate and molecular particles. When a molecule

impacts a surface, CTSP first determines the probability of the molecule striking the native material instead of another adsorbed molecule. This probability is given by $P = N_2\pi r^2/(\eta_0 A_{ele})$, where N_2 is the number of molecules adsorbed to the surface. CTSP then computes the residence time using equation 8 at the temperature of the impacted surface element. The molecule is re-emitted if $\tau_r/\Delta t \geq R$ where R is a random number. In this sense, $\tau_r/\Delta t$ is analogous to the sticking coefficient α_{sc} commonly utilized in other mass transport codes. For generality, CTSP allows the user to define a fixed α_{sc}, in which case the prescribed value will be used instead of the temperature-based model. For rebounding particles, we first sample the new velocity magnitude v_t from the Maxwellian speed distribution function at the surface temperature. The particle's new speed is set to $v + \alpha_a(v_t - v)$, where α_a is the thermal accommodation coefficient. Full accommodations $\alpha_a = 1$ is used in the examples presented in this paper. The new velocity direction follows the cosine law sampled according to a model of Greenwood[33]. On the other hand, if $\tau_r/\Delta t < R$ the particle is adsorbed to the surface layer. Computationally, this involves removing the particle from the simulation and incrementing the surface mass by $\Delta M_2 = w_{sp}m$. This term is correlated to the adsorption flux by $\Delta M_2 = \Gamma_a A_{ele}\Delta t$. For post-processing, the surface layer molecular mass can be converted to a film height by assuming spherical molecules,

$$h = \frac{M_2}{m}\frac{(4/3)\pi r^3}{A_{ele}} \tag{21}$$

A different model is used for particulates. Their post-impact velocity is given by a user-defined coefficient of restitution, $\alpha_r = v_2/v_1$. This parameter controls the "bounciness" of the particle. The particulate is allowed to leave until the post-impact velocity falls below some user defined threshold. The default value used in most of our simulations is 0.001 m/s. The mass is assumed to remain constant (i.e., the particle does not break up upon impact). Particulates without a sufficient post-impact velocity are deleted from the simulation and their cross-sectional area is used to update the target surface element percent area coverage,

$$\text{PAC} = 100\frac{\sum A_p}{A_{ele}} \tag{22}$$

PAC can then be converted to the corresponding cleanliness level at some slope C. Perry[29] provides a simple to use equation to map PAC to level. However, in experimenting with that model, we found the results to deviate by up to 5% for some values of C. Therefore, a precomputed lookup table is used to map PAC to a level.

V. Examples

We now demonstrate the code using four examples. The first one compares view factors between two parallel plates to the expected analytical prediction. The second example verifies the well-mixed outgassing model by computing the steady-state distribution of a contaminant inside a closed vessel. Next, the prescribed flux emission model is

demonstrated by simulating a commonly encountered engineering task: using a QCM to characterize the outgassing rate of a test article exposed to vacuum. The final example demonstrates the use of gaseous purge to reduce the infiltration of particulate contaminants.

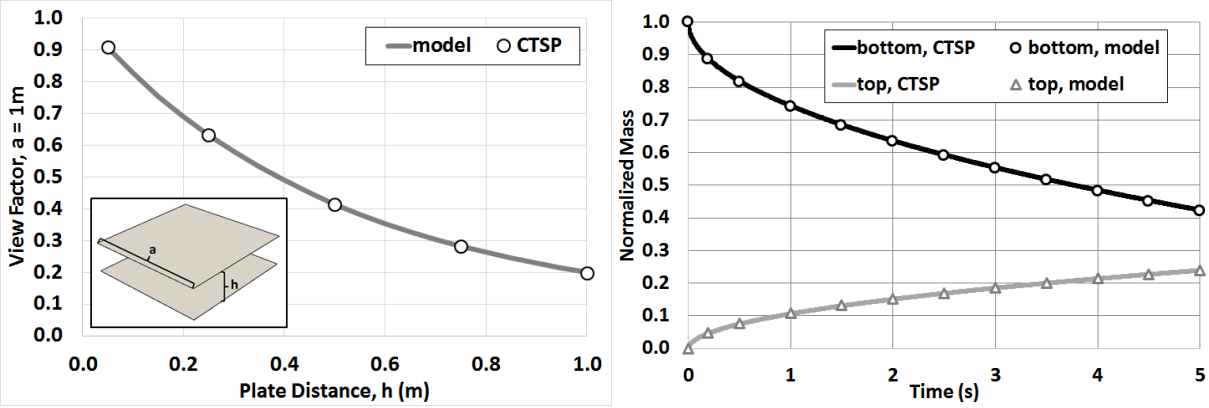
A. Example 1: View Factors

In the first example we consider a setup consisting of two identical square parallel plates with the edge length $a = 1$ m and separated by a distance h. This model can be used to test the CTSP particle injection algorithm. The analytical view factor between these two plates is given by[34]

$$k_{1\to 2} = \frac{2}{\pi w^2}\left[\ln\sqrt{1+\frac{w^4}{1+2w^2}} + 2w\left(\sqrt{1+w^2}\tan^{-1}\frac{w}{\sqrt{1+w^2}} - \tan^{-1} w\right)\right] \tag{23}$$

where $w = a/h$. This view factor can be computed numerically by injecting uniform flux from the bottom surface and setting the sticking coefficient on the top plate to 1. The ratio of the deposited to the emitted mass is the view factor. Molecules were injected for 1,000 time steps with $\Delta t = 2\times 10^{-4}$ s. The simulation then continued for an additional 1,000 time steps to assure that all emitted particles have either reached the top plate or have left the computational domain. Molecules were injected from random, spatially uniform positions with the initial velocity following the Lambertian distribution. The resulting view factor is compared to the analytical model for several different values of h in Figure 3a. An excellent agreement is seen, demonstrating that the surface emission model is implemented correctly. This particular run contained approximately 8,000 simulation molecules per time step.

We can also use this setup to compare the contamination modeling approach implemented in CTSP to the heritage approach used by the contamination community. The case with $h = 0.5$ m is considered. The bottom plate is assigned an initial $M_0 = 10^{-6}$ kg mass of some hypothetical contaminant having $E_a = 12$ kCal/mol, $m = 94$ amu, and $C_0 = 2.5\times 10^5$ kg/m^2/s$^{0.5}$ The power law outgassing model is used. The simulation is run with $w_{sp} = 10^{12}$, resulting in the number of particles present at each time step decreasing from around 28,000 to 800 over the duration of the simulation. The simulation is run for 25,000 $\Delta t = 2\times 10^{-4}$ s time steps, simulating 5 seconds of real time. This particular case took less than one minute to complete on a standard laptop computer. The mass outgassed from the bottom surface, normalized by the initial value M_0, is shown in Figure 3b. This plot also shows the normalized mass deposited on the top plate. These results can be compared to predictions from the heritage approach. Integrating Equation 3, we obtain $M_b(t) = M_0 - 2Ct^{0.5}A$ where $C = C_0\exp(-E_a/(RT))$, $A = 1$ m^2, and M_b is the mass remaining trapped within the bottom plate at time t. Predictions from the model are shown by the white circles. The triangles show the mass reaching the top plate, obtained by multiplying $M_0 - M_b(t)$ with the view factor $k_{1\to 2}$ from Equation 23. Again, an excellent agreement is seen between the CTSP and the model predictions. It should be noted that for an increasing distance between the plates, we would expect a small offset to develop due to the finite time needed for molecules to reach the



(a) View Factor

(b) Normalized Mass

Fig. 3 a) Simulation and analytical view factor for two square plates and b) normalized mass on the two plates as a function of time

target surface. The analytical model does not take this time of flight into account.

B. Example 2: Molecular Equilibrium

Consider a solid sphere placed inside a larger, hollow sphere. This closed system can be used to test various aspects of the molecular transport model. We let the inner sphere initially contain some molecular contaminant trapped within it but the rest of the system is contaminant free. Clearly, this system is not at equilibrium. The molecules present inside the sphere (region I) are expected to diffuse to the surface (region II) and then desorb into the gas phase (region III). These molecules will then impact the outer sphere and some fraction will deposit onto the outer sphere surface (region IV). These molecules then diffuse into the outer sphere bulk material (region V). At steady-state, concentration gradients between these five regions must vanish. For instance, from Equation 7, we have

$$-\frac{d\rho}{dt} \equiv H(\rho,\theta) = h(\rho - \gamma\theta) \tag{24}$$

so that at steady state $\rho = \gamma\theta$. We define $\rho = M_1/V_1$, where M_1 is the number of molecules inside the inner sphere of volume V_1. Similarly, $\theta = M_2/A_2$, with M_2 being the mass of molecules in the surface layer and A_2 is the surface area of the inner sphere. The equilibrium partition coefficient is set to $\gamma = 1$ m^{-1}. At steady state, the adsorption and desorption fluxes must also be equal, $\Gamma_a = \Gamma_d$.

We run this simulation with the radius of the inner sphere $r_1 = 10$ cm. The inner radius of the outer sphere is $r_2 = 1$ m, and the sphere is assumed to be 1 cm thick. The inner sphere contains 10^{-10} kg of the same hypothetical contaminant used in the previous example. The detailed model with $D_0 = 2.5 \times 10^{11}$ m^2/s is now used instead of the power-law approach. This combination of D_0, E_a, and T results in the diffusion coefficient $D \approx 8$ m^s/s. This value is over ten orders of magnitudes higher than actual diffusion coefficients for molecules in solids. It was selected to

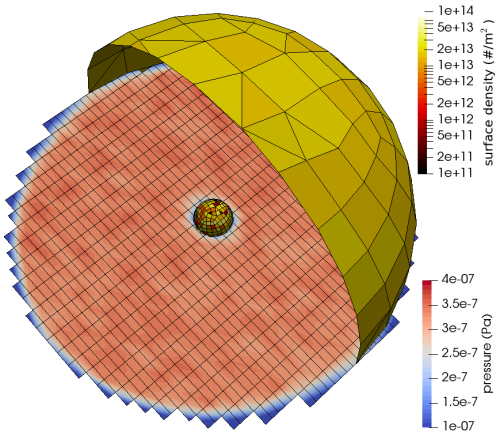


Fig. 4 Typical pressure and surface density for the isothermal case.

speed up the computation. The simulation is run for 1,000 $\Delta t = 10^{-4}$ s time steps, simulating 0.1 s of real time. The exact mass generation model for creating fractional weight particles is used, with the nominal w_{sp} set to 10^{10}. The simulation contained approximately 64,000 particles and took 1.5 minutes to run. We run this setup for two temperature configurations. The first assumes that the entire system is isothermal with both spheres at $T = 250$ K. The outer sphere temperature was reduced to 200 K in the second configuration.

Figure 4 visualizes the setup and also plots the typical contaminant partial pressure inside the gas cavity for the isothermal case. As expected, the pressure is uniform. This field was generated by scattering simulation particle positions and velocities to the grid to compute the node-averaged number density and temperature. The ideal gas law is then used to compute pressure. Surface molecular number density is also plotted. We can notice an increased noise level on the inner sphere due to the smaller surface elements receiving on average fewer simulation particles per element. This noise could be reduced by running with more simulation particles or by using a coarser mesh. Next, Figure 5 shows the evolution of the mass contained within each of the five regions. Initially, all mass is contained inside the inner sphere, shown by the thick black line. This value then rapidly decays until it reaches an equilibrium around $t = 0.06$ s. Note that this rapid decay is a direct product of the artificially high diffusion coefficient used in this example. Real systems require hours or even days to reach a similar equilibrium at comparable temperatures. The loss of mass from the inner sphere bulk is accompanied by mass increase in the other regions, including the inner sphere surface (dashed black line), the gas phase (dash-dotted light gray line), the outer sphere surface (dashed dark gray line), and the outer sphere bulk material (solid dark gray line). At steady state, majority of mass is found in the gas phase and on the outer sphere surface. These two regions have the largest spatial dimensions. The total mass, shown by the dotted line, remains constant over the entire simulation.

The corresponding mass densities are shown in Figure 6a. Densities for Regions I, III, and V are obtained by dividing the contained mass by volumes of the inner sphere V_1, the gas region V_3, and the outer sphere V_5. Similarly,

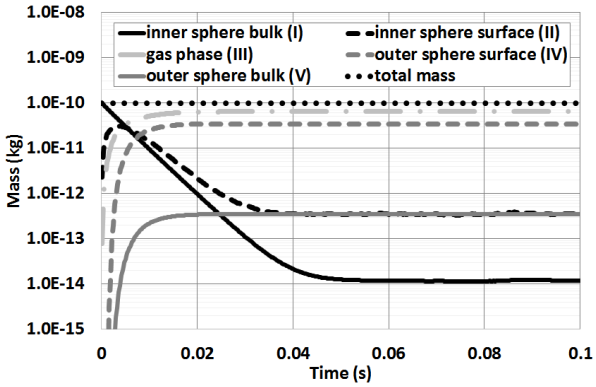
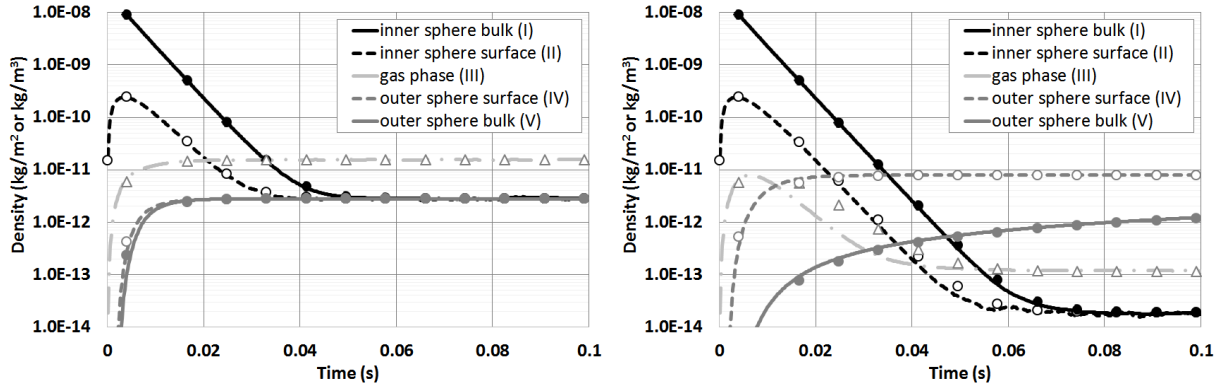


Fig. 5 Molecular concentration and molecular mass for the isothermal spheres case



(a) Isothermal

(b) Cold Outer Sphere

Fig. 6 Molecular mass density for the a) isothermal and b) 200 K outer sphere case. Symbols show results from the analytical model.

the surface mass density on Regions II and IV is obtained by dividing the deposited mass by the respective surface areas A_2 and A_4. We can note that at steady state, the concentration on both spheres are identical. This finding is expected, yet is still interesting, given that the two spheres are not in a physical contact with each other. Figure 6b shows the corresponding plot for the case with the outer sphere temperature reduced to 200 K. This time, the system is not yet in a full equilibrium, due to the surface concentration of the outer sphere (Region IV) exceeding the density within the sphere (Region V). The equilibrium could be achieved by running the simulation for an additional 1.5 s. The mass concentration on the outer sphere surface is now greatly increased compared to the isothermal case, while the concentration in the gas phase is greatly decreased. This result is also expected, since the colder outer sphere acts as a sink for the molecular contaminants.

The results from the kinetic CTSP simulation are also compared to an analytical model. We let the mass evolution

in the five regions be governed by the following mass conservation statements:

$$\begin{aligned}
dM_1/dt = &\quad (-dM_{12} + dM_{21}) \\
dM_2/dt = &\quad (+dM_{12} - dM_{21}) + \quad (-dM_{23} + dM_{32}) \\
dM_3/dt = &\qquad\qquad\quad (+dM_{23} - dM_{32}) + \quad (-dM_{34} + dM_{43}) \\
dM_4/dt = &\qquad\qquad\qquad\qquad\quad (+dM_{34} - dM_{43}) + \quad (-dM_{45} + dM_{54}) \\
dM_5/dt = &\qquad\qquad\qquad\qquad\qquad\qquad\quad (+dM_{45} - dM_{54})
\end{aligned} \tag{25}$$

where

$$\begin{aligned}
dM_{12} &= h_1(M_1/V_1)A_2\Delta t \\
dM_{21} &= h_1\gamma_1 M_2\Delta t \\
dM_{23} &= M_2/\tau_{r2}\Delta t \\
dM_{32} &= (M_3/V_3)(\bar{u}/4)A_2\Delta t \\
dM_{34} &= (M_3/V_3)(\bar{u}/4)A_4\Delta t \\
dM_{43} &= M_4/\tau_{r4}\Delta t \\
dM_{45} &= h_5 * \gamma_4 * M_4\Delta t \\
dM_{54} &= h_5(M_5/V_5)A_4\Delta t
\end{aligned} \tag{26}$$

Here h_1 and h_5 are the scaled diffusion coefficients for the inner and the outer sphere. The residence times on surfaces of these these two regions are given by τ_{r2} and τ_{r4}. The term dM_{12} correspond to the mass diffusing from the inner sphere bulk to the inner sphere surface while dM_{21} is the reverse mass diffusion from the surface to the bulk. These relationships are obtained directly from Equation 7. The mass desorption from the surface to the gas phase is given by dM_{23}. This term equals to $\Gamma_d A_2 = (\theta_2/\tau_r)A_2$, since $\theta_2 A_2 = M_2$. The terms dM_{32} and dM_{34} capture the mass flow from the gas phase onto the inner and the outer sphere, respectively. From kinetic theory, the surface impingement flux is given by $\Gamma_a = (M_4/V_3)\bar{u}/4$, where the term in parentheses is the mass density, and the mean velocity $\bar{u} = \sqrt{8kT/(\pi m)}$. The desorption flux is $\Gamma_d = \theta/\tau_r$. The above scheme is integrated numerically using the Forward Euler method. The resulting time dependent values of M_1/V_1, M_2/A_2, M_3/V_3, M_4/A_4, and M_5/V_5 are plotted by the symbols in Figure 6a. An excellent agreement is seen despite CTSP directly implementing only a subset of the above equations. Specifically, the deposition from the gas phase onto the surface is modeled by particles, which individually have no concept of a mean velocity or the gas mass density. CTSP also directly re-emits particles with probability $1 - \tau_r/\Delta t$.

The masses at the end of the simulation can be used to compute fluxes,

$$\begin{aligned}
\Gamma_{12} &= h_1 M_1/V_1 = 2.269 \times 10^{-11} \\
\Gamma_{21} &= h_1 \gamma_2 M_2/A_2 = 2.269 \times 10^{-11} \\
\Gamma_{23} &= M_2/A_2/\tau_{r2} = 9.076 \times 10^{-10} \\
\Gamma_{32} &= (M_3/V_3)\bar{u}/4 = 9.076 \times 10^{-10} \\
\Gamma_{34} &= (M_3/V_3)\bar{u}/4 = 9.076 \times 10^{-10} \\
\Gamma_{43} &= M_4/A_4/\tau_{r4} = 9.076 \times 10^{-11} \\
\Gamma_{45} &= h_5 \gamma_4 M_4/A_4 = 2.269 \times 10^{-11} \\
\Gamma_{54} &= h_5 M_5/V_5 = 2.269 \times 10^{-11}
\end{aligned} \tag{27}$$

Again, the system can be seen to be in equilibrium. The surface desorption flux Γ_{23} and Γ_{43} equal to the random thermal flux predicted by the kinetic theory, Γ_{32} and Γ_{34}. The diffusive fluxes between the bulk material and the surface Γ_{12} and Γ_{21} also equal. The above model can also be run for the case with the colder outer sphere from Figure 6b. While the model still predicts the identical steady-state as the CTSP simulation, the time evolution of the gas phase concentration in Region III differs. This discrepancy likely arises from the analytical model assuming $T = (T_2 A_2 + T_4 A_4)/(A_2 + A_4) \approx 200.5$ K for the gas phase (in order to compute $\bar{u}$). This temperature is not achieved until the steady state is reached as it requires that molecules re-emitted from the outer sphere propagate back through the simulation domain.

C. Example 3: Outgassing Rate

We next consider a commonly encountered engineering task: characterizing the outgassing rate of some test article exposed to vacuum. A device known as the Quartz Crystal Microbalance (QCM)[15] can be used for this purpose. This instrument operates by exposing a crystal to the vacuum environment. The vibrational frequency of the crystal is proportional to the amount of mass deposited on it. A secondary unexposed crystal provides a reference baseline to account for thermal drift. A Thermoelectric QCM (TQCM) also contains a temperature control system capable of maintaining the crystal at some preset value. During an outgassing characterization test, a QCM is used to measure the rate with which contaminants deposit on the crystal. This deposition rate then needs to be correlated to the outgassing rate of the test hardware. Occasionally it may be possible to position the QCM in front of the test article and obtain this measurement directly. That approach is not practical for the large vacuum facilities used in thermal vacuum testing, which may contain a QCM mounted in the chamber wall meters away from the spacecraft. Mass conservation then needs to be considered. At steady state, the mass production rate from the outgassing process must be balanced by the

collection rate at the pumps and other cold sink surfaces. If we assume that the chamber is in a *hot wall* configuration in which the chamber walls and any supporting structures are too warm for the contaminants to condense, we have

$$\Gamma_h A_h = \Gamma_q A_q - \Gamma_p A_p \tag{28}$$

where $\Gamma_h \equiv \dot{m}_h$ is the outgassing mass flow rate (kg/s) of the hardware given in terms of the surface flux (kg/m^2/s) and the test article surface area (m^2). We have similar terms on the right hand side for the QCM and the pumps. If the QCM and pumps are equally cold sinks, $\Gamma_q = \Gamma_p$, and the above equation simplifies to

$$\Gamma_h A_h = \Gamma_q (A_q + A_p) \tag{29}$$

Alternatively, we can write $\Gamma_h k_{h\to q} = \Gamma_q$, where $k_{h\to q} = A_h/(A_q + A_p)$ is the view factor from the test article to the QCM. For large chambers, we can take $\dot{m}_h = \Gamma_q A_p$ since $A_p \gg A_q$. From this relationship we can see that by knowing the pump area A_p, we can compute the outgassing rate from the QCM deposition mass flux, Γ_q. Although this calculation appears trivial, the difficulty arises from the A_p term. Simply considering the cross-sectional area of the pumps would ignore pumping inefficiencies and conductance losses through the connecting ducts[35]. Instead, we can experimentally determine the effective pump area following the approach outlined in the ASTM-E2900 standard and temporarily introduce a secondary sink of a known area, such as a scavenger plate flooded with liquid nitrogen[36]. We can then write two mass conservation equations, one without and one with the scavenger plate,

$$\Gamma_\mathrm{h} \mathrm{A_h} = (\Gamma_\mathrm{q})_1 (\mathrm{A_q} + \mathrm{A_p}) \tag{30}$$

$$\Gamma_\mathrm{g} \mathrm{A_g} = (\Gamma_\mathrm{q})_2 (\mathrm{A_q} + \mathrm{A_p} + \mathrm{A_s}) \tag{31}$$

The hardware outgassing term can be eliminated, leading to

$$A_p = \frac{(\Gamma_q)_2}{(\Gamma_q)_1 - (\Gamma_q)_2} A_s - A_q \tag{32}$$

The computed A_p is then substituted into Equation 29 to obtain the hardware outgassing rate.

This approach is demonstrated numerically. We consider a setup consisting of a harness placed inside a bell jar, as shown in Figure 7. The harness is placed on a platten which partly blocks the entrance to a long, curved duct leading to a pump. A QCM, with dimensions based on the commonly used QCM Research Mark-10 is placed near the harness. The platten also supports a scavenger plate.

We prescribe a constant $\Gamma_a = 10^{-8}$ kg/m^2/s flux on the harness. The objective of the simulation is to recover this flux

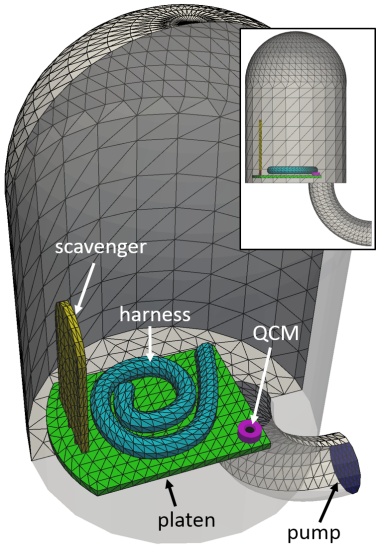
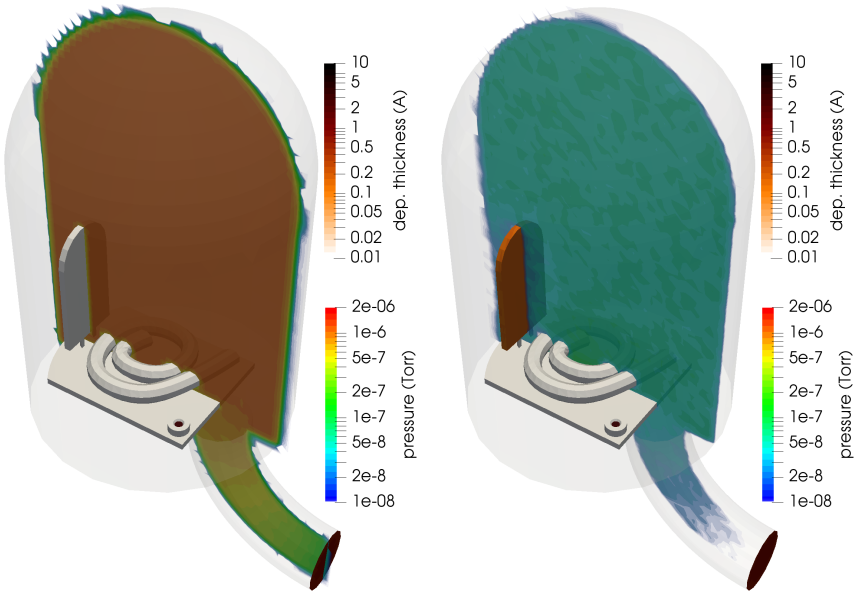


Fig. 7 Simulation geometry with the major components identified.

from the mass deposition rate on the QCM. The harness outgasses the same hypothetical 94 amu, $E_a = 12$ kCal/mol, $r = 1.55 \times 10^{-10}$ m contaminant used previously. The stochastic mass generation model is used to inject simulation particles with the specific weight $w_{sp} = 5 \times 10^{10}$. The simulation runs for 200,000 $\Delta t = 2 \times 10^{-4}$ s time steps, therefore simulating 40 s of real time. CTSP's ability to dynamically alter component temperatures is used to drop the scavenger temperature from 300 K at $t = 15$ s to 15 K at $t = 15.5$ s. All other surfaces remain at 300 K, with the exception of the pump and the QCM crystal, which are set to 15 K. It should be noted that TQCMs normally do not operate at such cryogenic temperatures but this simplification was taken to assure that $\Gamma_q = \Gamma_p$.

Figure 8a shows the contaminant partial pressure just prior to the scavenger activation. The data for this plot was obtained by averaging 2,000 instantaneous samples taken every at five time step interval to smooth out the numerical noise. The surface thickness of the deposited molecular film, computed from Equation 21, is also shown. At first, only the QCM crystal and the pump are cold enough to collect the contaminant and hence are the only surfaces with a non-zero deposition thickness. The steady-state achieved with the scavenger activated near the end of the simulation is plotted in Figure 8b. The contaminant partial pressure decreases by over an order of magnitude from 1.1×10^{-6} Torr to 5×10^{-8} Torr. This pressure drop is expected due to the increase in the total sink area. The scavenger now also contains some finite mass deposited on it.

The total mass deposited onto the QCM crystal over the duration of the simulation can be seen in Figure 9. This Figure also plots the total number of simulation macroparticles. We can clearly see the impact of the scavenger activation. The number of macroparticles drops from around 46,000 with the warm scavenger to approximately 2,000 with the scavenger activated. Since each macroparticle corresponds to the same number of real molecules, this change is directly proportional to a decrease in the contaminant number density and hence pressure. We can also clearly see a difference



(a) t = 15 s (b) t = 40 s

Fig. 8 Contaminant partial pressure before (a) and after (b) scavenger activation. Contaminant surface layer height is also shown.

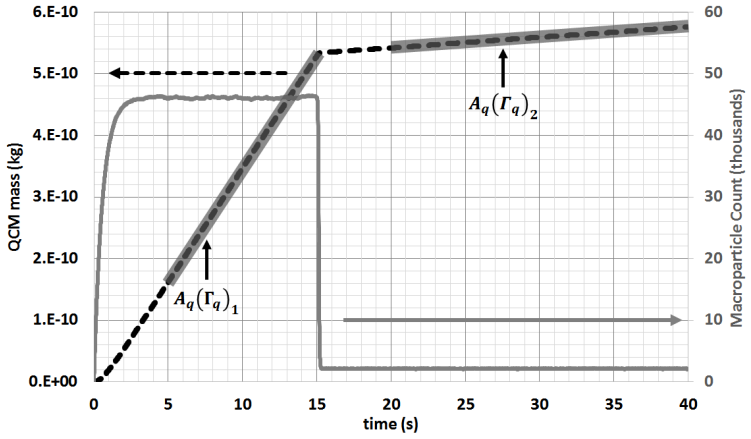


Fig. 9 Mass collected by the QCM. Difference in the mass deposition rate before and after the scavenger plate activation is apparent. The gray line plots the number of simulation macroparticles, in thousands.

Simulation Inputs		Computed Values	
A_q	1.11×10^{-4} m^2	$A_q(\Gamma_q)_1$	3.66×10^{-11} kg/m^2/s
A_s	4.23×10^{-2} m^2	$A_q(\Gamma_q)_2$	1.71×10^{-12} kg/m^2/s
$(A_p)_{geom}$	7.63×10^{-3} m^2	$(A_p)_{eff}$	1.96×10^{-3} m^2
A_h	6.24×10^{-2} m^2	$(\Gamma_h)_{QCM}$	1.10×10^{-8} kg/m^2/s
		error	9.9 %

Table 1 Summary of simulation inputs and results for the bell jar example.

case	A_s (m^2)	Γ_q (kg/m^2/s)	Γ_s (kg/m^2/s)	error (%)	$(A_s)_{eff}/A_s$ (%)
I (f/b/s)	4.23×10^{-2}	1.54×10^{-8}	1.41×10^{-8}	8.68	91.32
II (f/b)	3.77×10^{-2}	1.73×10^{-8}	1.57×10^{-8}	9.55	90.45
III (f)	1.88×10^{-2}	2.74×10^{-8}	3.03×10^{-8}	-10.44	110.44
IV (b)	1.88×10^{-2}	4.60×10^{-8}	2.85×10^{-8}	38.01	61.99

Table 2 Characterization of the scavenger plate effective area. The scavenger and QCM fluxes were assumed to remain equal regardless of the scavenger configuration, which is not the case.

in the QCM mass deposition rate (the slope of the mass curve) before and after the scavenger activation at $t = 15$ s. A linear fit can be made to the two highlighted regions in $t = [5, 15]$ s and $t = [20, 40]$ s. The slopes of these two segments correspond to $A_q(\Gamma_q)_1$ and $A_q(\Gamma_q)_2$, and their ratio is used to compute the pump area per Equation 32. The inputs and the computed results are summarized in Table 1. The geometric areas of the pump, the scavenger, and the QCM are obtained by summing the areas of surface mesh triangles or quadrangles making up the respective components. The computed effective pump area, $A_p = 1.96 \times 10^{-3}$ is found to be 25 % of the geometric pump area. Substituting this value into Equation 29, we obtain the harness outgassing flux

$$\Gamma_a = (\Gamma_q)_1 \frac{A_p + A_q}{A_h} = 1.10 \times 10^{-8} \text{ kg/m}^2\text{/s} \tag{33}$$

which is 10% higher than expected. While this overestimation would be well within bounds of a typical experimental margin of error, it is nevertheless surprising to encounter it in a numerical simulation of a closed system. The error thus warrants a further investigation. The usual culprit in kinetic simulations is the numerical noise associated with a finite number of computational particles. Instead of running a case with more particles by reducing w_{sp}, we can instead re-run this simulation several times to obtain an insight into the statistical variation between runs. Five additional runs were completed, resulting in $\overline{A_q(\Gamma_q)_1} = 3.65 \times 10^{-11} \pm 0.26\%$ kg/m^2/s, $\overline{A_q(\Gamma_q)_2} = 1.69 \times 10^{-12} \pm 1.54\%$ kg/m^2/s, $\overline{A_p} = 1.94 \times 10^{-3} \pm 1.56\%$ m^2, and $\overline{\Gamma_h} = 1.10 \times 10^{-8} \pm 1.61\%$ kg/m^2/s. The error margins correspond to one standard deviation normalized by the mean value. The resulting average hardware outgassing flux overestimates by expected values by 8.3%, which corresponds to over five standard deviations. Clearly, this difference cannot be explained by numerical noise alone.

Instead, the discrepancy arises from an error in our guiding assumptions. In writing Equation 31, we assumed that $\Gamma_s = (\Gamma_q)_2$. Besides implying that both surfaces are equally cold, this simplification also assumes that the probability of a molecule originating from the harness impacting the scavenger plate is given by the ratio of the scavenger surface area to the total sink area. This suggests that there should not be a direct line of sight from the harness to the scavenger. At the same time, there should not exist a torturous path to reach the scavenger. Consider the limiting case in which the scavenger completely covers the test article. The entire mass outgassed by the harness will be collected by the scavenger

and no molecules will reach the QCM or the pump. On the other hand, a scavenger located behind a long duct will suffer from similar conductance losses previously observed with the effective pump area. Similar geometric effects are found here. Due to their close proximity, the harness has a finite direct line of sight view of the scavenger. Conversely, the scavenger is placed near the wall and therefore the pumping on the back face is limited to the regions near the edges that are easier for molecules to reach. Per the assumption in Equation 31, the mass collected by the scavenger should scale according to $\dot{m}_s = A_s(\Gamma_q)_2$. The model used previously contained three active regions: the front face facing the harness, the equally-sized back face near the wall, and a C-shaped side wall. We can investigate this scaling relationship by running several additional simulations with a varying configuration of the active surfaces: the three above mentioned regions (I), the front and the back face without the side wall (II), the front face only (III), and the back face only (IV). In order to reduce numerical noise, this set of simulations was run with w_{sp} decreased two-fold to 2.5×10^{10} and the scavenger kept cold over the entire duration of the simulation. This allows us to compute $\dot{m}_s$ and $(\Gamma_q)_2$ using the longer $t = [5, 40]$ time interval. The results are summarized in Table 2. Clearly, the assumption that $\Gamma_s = \Gamma_q$ regardless of the actual active region design does not hold. This particular scavenger plate acts as a model cold surface with an effective area ranging from 62 to 110% of the geometric value. The resulting 8.7% error observed for Case I agrees with the mean error obtained from the six simulation runs. The majority of the discrepancy between the simulation and prescribed harness outgassing flux can thus be attributed to an erroneous value for the effective scavenger area. Cases III and IV compare the flux to the front and the back face. Per the above discussion, the harness-facing front face acts as a cold plate with an increased effective area due to the direct line of sight effects. On the opposite side, only 62% of the total geometric area is active due to the close proximity of the surface to the wall. This study indicates that the placement and the design of the scavenger plate is of importance. Effective pump area characterizations are generally made in an empty chamber. The material diffusing from the chamber walls acts as an isotropic source helping to reduce these line-of-sight inaccuracies.

D. Example 4: Particulate Contamination

In the final example we consider the effect of nitrogen purge on particulate contamination in atmospheric conditions. Suppose that some instrument containing two detectors is stored on a lab bench. The instrument is connected to a nitrogen purge which vents out around the perimeter of the larger of the two detectors. An impact to the bench dislodges particulates from the shelf above the detector. We are interested in numerically studying the effectives of the purge in preventing these particulates from reaching the detector. The entire set up can be seen in Figure 10. As CTSP does not contain a built-in fluid solver, the flow profile due to the purge gas needs to be computed externally. The Simflow, front-end for the OpenFOAM, solver[37] was used for this purpose. Five flow rates were considered: 0, 1×10^{-4}, 2×10^{-4}, 5×10^{-4} and 1×10^{-3} m^3/s. Each simulation assumed initial cleanliness level $L = 600$ with $C = 0.926$ on the bottom surface of the top shelf. The Klavins and Lee detachment model with $a = 4$ g was used to simulate the impact.

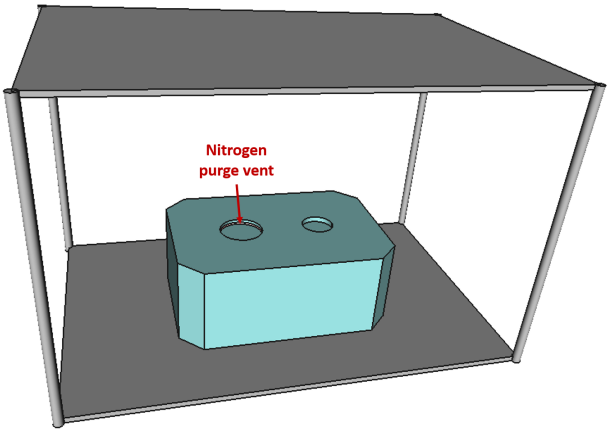
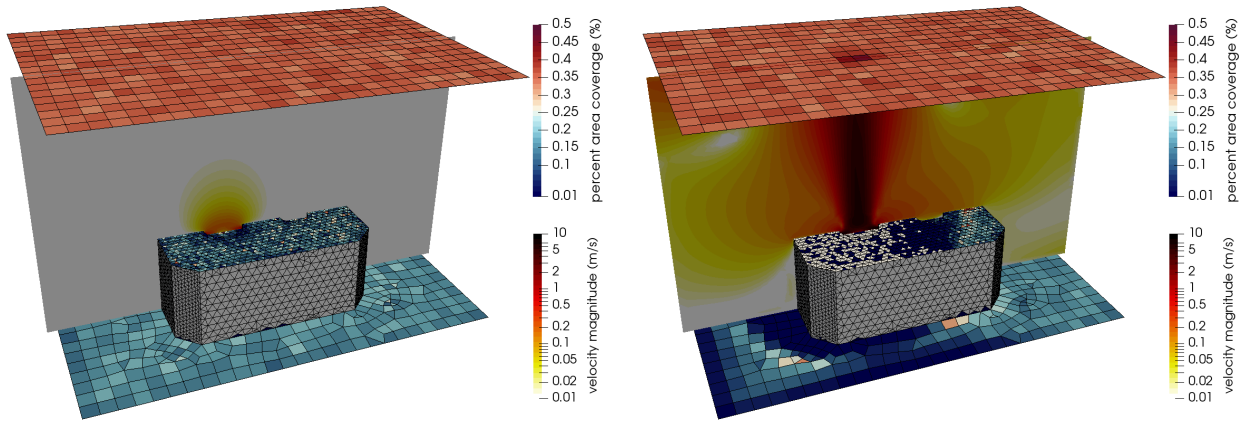


Fig. 10 Model setup for the particulate redistribution example. Gaseous nitrogen purge vents out around the perimeter of the larger detector.

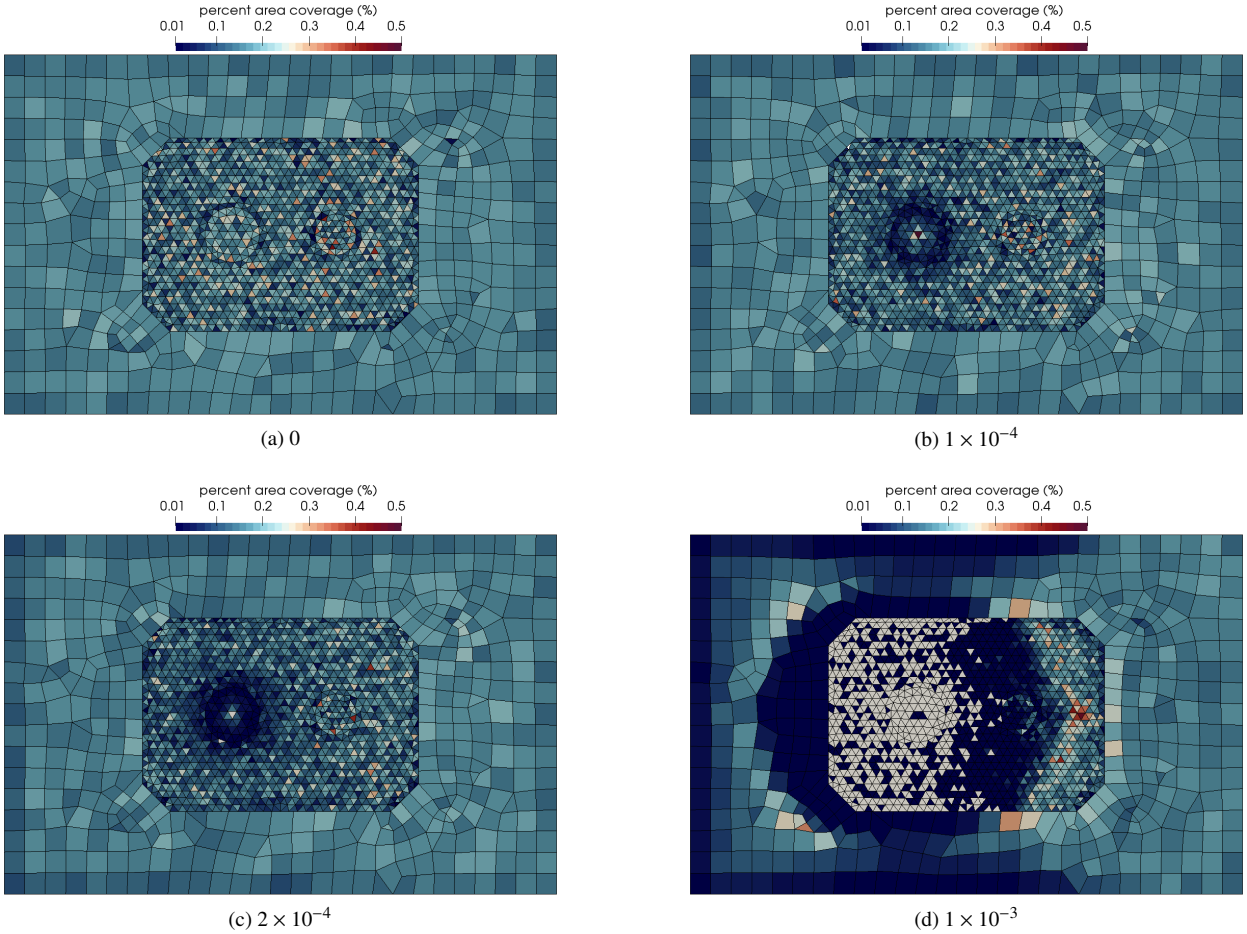


(a) 1×10^{-4} m^3/s (b) 1×10^{-3} m^3/s

Fig. 11 Visualization of flow profiles and the corresponding surface percent area coverage for two values of inlet velocity.

The standard 1 g gravity was applied. Figure 11 visualizes the flow profiles for the $Q = 1 \times 10^{-4}$ and $Q = 1 \times 10^{-3}$ m^3/s purge rates. The shelves were meshed using a single layer of elements and hence the contouring on the top shelf corresponds to particulate loading on the bottom surface. We can clearly see that an increase in purge flow results in a decreased particulate concentration on the instrument. The ≈ 0.35 PAC on the top shelf corresponds to the particulates unable to detach under the 4 g impact.

The impact of purge is further visualized in Figure 12. These plots show the top-down view of the instrument. We can note that in the case with no flow, the PAC on the instrument top surface is comparable to the PAC on the bottom shelf despite differences in mesh element sizes. This observation helps us confirm that the PAC computation is mesh independent. The particulate concentration on the detector decreases as the purge flow is increased. At $Q = 1 \times 10^{-3}$ m^3/s, the flow rate is sufficiently high to act as an umbrella, reducing deposition even on the neighbor detector without



(a) 0

(b) 1×10^{-4}

(c) 2×10^{-4}

(d) 1×10^{-3}

Fig. 12 Variation in particulate percent area coverage with the purge gas flow rate.

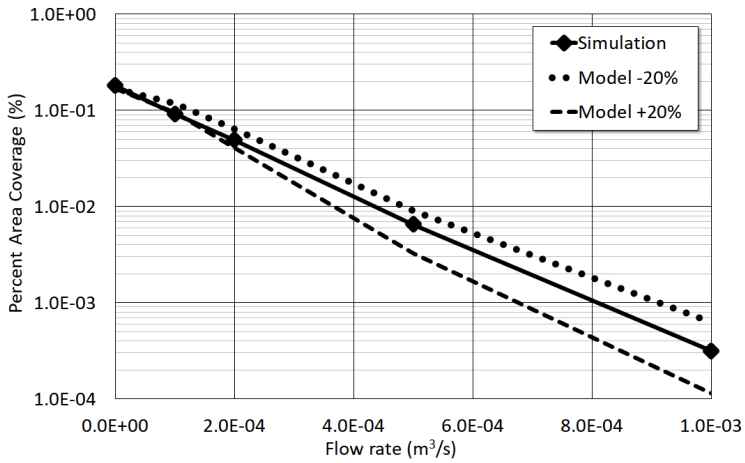


Fig. 13 Comparison of average PAC on the detector to predictions from a Python model run with ±20% mean axial velocity.

the purge vent. Due to the complexity of this setup, it is difficult to come up with an exact analytical model. Instead, a simplified numerical model was developed in Python. The model computes the number of particulates released from a surface given an initial cleanliness level, slope, and impact acceleration. The released particulate counts are computed in 1 μm increments in the range of 1 to 1000 μm. Particulate sizes and cross-sectional areas are set using the aspect ratios from Equation 12. The gravitational force $F_g = mg$ is then compared to the drag force $F_d = 0.5\rho C_d A u_z^2$ for some input value of axial flow velocity u_z for each 1 μm bin. If $F_g > F_d$, all particulates in that bin are assumed to reach the bottom surface, and their total area is used to update the target PAC. This model requires that a value of u_z is specified for each flow rate. This parameter is the major source of uncertainty due to the non-uniformity in the velocity flow profile, as seen in Figure 11. Paraview's histogram filter was used to obtain this value in a consistent manner for all five cases. The CFD solution data points were selected on the plane of symmetry over a rectangular region where the flow velocity was non-negligible. The mean "x" axis of the histogram was used to set the velocity. The sampled values were 0, 0.098, 0.295, 1.136, and 2.7 m/s, respectively. The comparison of the model predictions to the simulation results in shown in Figure 13. The dashed and dotted lines plot the model predictions for $u_z \pm 20\%$. An excellent agreement is seen.

VI. Conclusion

A new simulation program for modeling transport of molecular and particulate contaminants in free-molecular, transition, and continuum regimes is described. CTSP concurrently traces multiple particles through small simulation time steps. At each time step, particle velocities are updated based on user-provided force fields. The concurrent push allows the code to compute macroscopic properties such as contaminant plume number density or vacuum chamber pressure. CTSP implements multiple material source models applicable to contamination modeling, including a detailed model for molecular outgassing and particulate redistribution. The code is illustrated with four examples. The first

example compares simulated view factors to an analytical model. The second example then confirms that equilibrium is achieved at steady state for a closed system. Next, a common engineering task of characterizing the outgassing rate of a test article using a QCM is simulated. The final example models the use of gaseous purge to reduce particulate contamination. All examples showed an excellent agreement with analytical or reduced models. Future effort on the code includes additional code validation, ideally using experimental data. Furthermore, code parallelization and optimization is currently in progress. Finally, a first-principle model of particle detachment using the Discrete Element Method is under investigation.

Acknowledgments

The author would like to acknowledge many fruitful discussions with members of the contamination control community including Mike Woronowicz, Tim Gordon, Robert Studer, Therese Errigo, David Hughes, and De-Ling Liu. This work was supported by internal R&D funds.

References

[1] Colony, J. A., "Ultraviolet Absorption of Common Spacecraft Contaminants," Tech. Rep. NASA TM 80551, Goddard Space Flight Center, 1979.

[2] Tribble, A., *Fundamentals of Contamination Control*, SPIE, 2000.

[3] Dever, J. A., "Low Earth Orbital Atomic Oxygen and Ultraviolet Radiation Effects on Polymers," Tech. Rep. NASA TM 103711, Lewis Research Center, 1991.

[4] Stewart, T., Arnold, G. S., Hall, D., D.C., M., Hwang W.C, R., Young Owl, and H.D., M., "Photochemical Spacecraft Self-Contamination: Laboratory Results and Systems Impacts," Tech. Rep. AD-A226 488, Aerospace Corp, 1990.

[5] Galica, G. E., Green, B. D., Boies, M. T., Uy, O. M., Silver, D. M., Benson, R. C., Erlandson, R. E., Wood, B. E., and Hall, D. F., "Particle Environment Surrounding the Midcourse Space Experiment Spacecraft," *Journal of Spacecraft and Rockets*, Vol. 36, No. 4, 1999.

[6] O'Dell, S. L., Swartz, D. A., Tice, N. W., Plucinsky, P. P., Grant, C. E., Marshall, H. L., Vikhlinin, A., and Tennant, A. F., "Modeling contamination migration on the Chandra X-ray Observatory II," *Optical Engineering and Applications*, International Society for Optics and Photonics, 2013, pp. 88590F–88590F.

[7] Cauffman, D., "Ionization and Attraction of Neutral Molecules to a Charged Spacecraft," Tech. Rep. SD-TR-80-78, The Aerospace Sciences Corp., 1980.

[8] Scialdone, J. J., "Self Contamination and Environment of an Orbiting Spacecraft," Tech. Rep. NASA TN D-6645, Goddard Space Flight Center, 1972.

[9] Hastings, D., and Garrett, H., *Spacecraft-Environment Interactions*, Cambridge University Press, 1996.

[10] Lai, S., "Spherical-shaped ice particle production by spraying water in a vacuum chamber," *IEEE Transaction on Nuclear Science*, Vol. 36, No. 6, 1989, pp. 2027–2032.

[11] Chang, C. W., Kannenberg, K., and Chidester, M. H., "Development of versatile molecular transport model for modeling spacecraft contamination," *Optical Systems Contamination: Effects, Measurements, and Control*, Vol. 7794, International Society for Optics and Photonics, 2010.

[12] Rantanen, R., and Gordon, T., "Electrostatic Return of Contaminants," Tech. Rep. NASA/CR-2003-212637, Prepared for Marshall Space Flight Center, 2003.

[13] Brieda, L., Barrie, A., Hughes, D., and Errigo, T., "Analysis of particulate contamination during launch of the MMS mission," *Optical System Contamination: Effects, Measurements, and Control*, Vol. 7794, International Society for Optics and Photonics, 2010. 77940P.

[14] LeBeau, G., and Lumpkin, F. E., "Application highlights of the DSMC Analysis Code (DAC) software for simulating rarefied flows," *Computer Methods in Applied Mechanis and Engineering*, Vol. 191, No. 6-7, 2001, pp. 595–609.

[15] O'Sullivan, C., and Guilbault, G., "Commercial quartz crystal microbalances - theory and applications," *Biosensors and Bioelectronics*, Vol. 14, No. 8-9, 1999, pp. 663–670.

[16] Spektor, R., Matlock, T., M.W., C., D.T., S., and Brieda, L., "Analytical Pumping Speed Models for Electric Propulsion Vacuum Facilities," *Space Propulsion Conference*, Asociation Aeronautique et Astronautique de France, 2018.

[17] Bird, G., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Science Publications, 1994.

[18] Birdsall, C., and Langdon, A., *Plasma physics via computer simulation*, Institute of Physics Publishing, 2000.

[19] Koo, J., "Hybrid PIC-MCC Computational Modeling of Hall Thrusters," Ph.D. thesis, Univesity of Michigan, 2005.

[20] Shin, H., Lee, Y., and Jurng, J., "Spherical-shaped ice particle production by spraying water in a vacuum chamber," *Applied Thermal Engineering*, Vol. 20, 2000, pp. 439–454.

[21] Woronowicz, M. S., "Highlights of Transient Plume Impingement Model Validation and Applications," *42nd AIAA Thermophysics Conference*, 2011.

[22] Fang, W., Shillor, M., Stahel, E., Epstein, E., Ly, C., J., M., and Zaron, E., "A Mathematical Model for Outgassing And Contamination," *SIAM Journal of Applied Math*, Vol. 51, No. 5, 1991.

[23] ASTM International, *Standard Test Method for Contamination Outgassing Characteristics of Spacecraft Materials*, ASTM-E1559, 2009.

[24] Crank, J., *The Mathematics of Diffusion*, 2nd ed., Oxford Science Publications, 1980.

[25] Tribble, A. C., Boyadjian, B., Davis, J., Haffner, J., and McCullough, E., "Contamination control engineering design guidelines for the aerospace community," Tech. Rep. NASA Contractor Report 4740, NASA, 1996.

[26] IEST, *Product Cleanliness Levels - Applications, Requirements, and Determination*, IEST-STD-CC1246D, 2002.

[27] Raab, J., "Particulate Contamination Effects on Solar Cell Performance," Tech. rep., 1986.

[28] Ma, P., Fong, M., and Lee, A., "Surface Particle Obscuration and BRDF Predictions," *Scatter from Optical Components*, Vol. 1165, International Society for Optics and Photonics, 1989, pp. 381–391.

[29] Perry, R., "A Numerical Evaluation of the Correlation of Surface Cleanliness Level and Percent Area Coverage," *Optics and Photonics*, International Society for Optics and Photonics, 2006.

[30] Klavins, A., and Lee, A., "Spacecraft particulate contaminant redistribution," 1987, pp. 236–244.

[31] White, F., *Viscous Fluid Flow*, McGraw-Hill, New York, 1991.

[32] Brieda, L., "Numerical Investigation of Chamber Repressurization," *NASA Goddard Contamination, Coatings,Materials, and Planetary Protection Workshop*, 2017.

[33] Greenwood, J., "The correct and incorrect generation of cosine distribution of scattered particles for Monte-Carlo modelling of vacuum systems," *Vacuum*, Vol. 67, 2002, pp. 217–222.

[34] Howell, J., Menguc, M. P., and Siegel, R., *Thermal Radiation Heat Transfer*, CRC Press, 2015.

[35] O'Hanlon, J., *User's Guide to Vacuum Technology*, Wiley & Sons, 2003.

[36] ASTM International, *Standard Practice for Spacecraft Hardware Thermal Vacuum Bakeout*, ASTM-E2900, 2012.

[37] Jasak, H., Jemcov, A., and Tukovic, Z., "OpenFOAM: A C++ Library for Complex Physics Simulations," *International Workshop on Coupled Methods in Numerical Dynamics*, 2007.