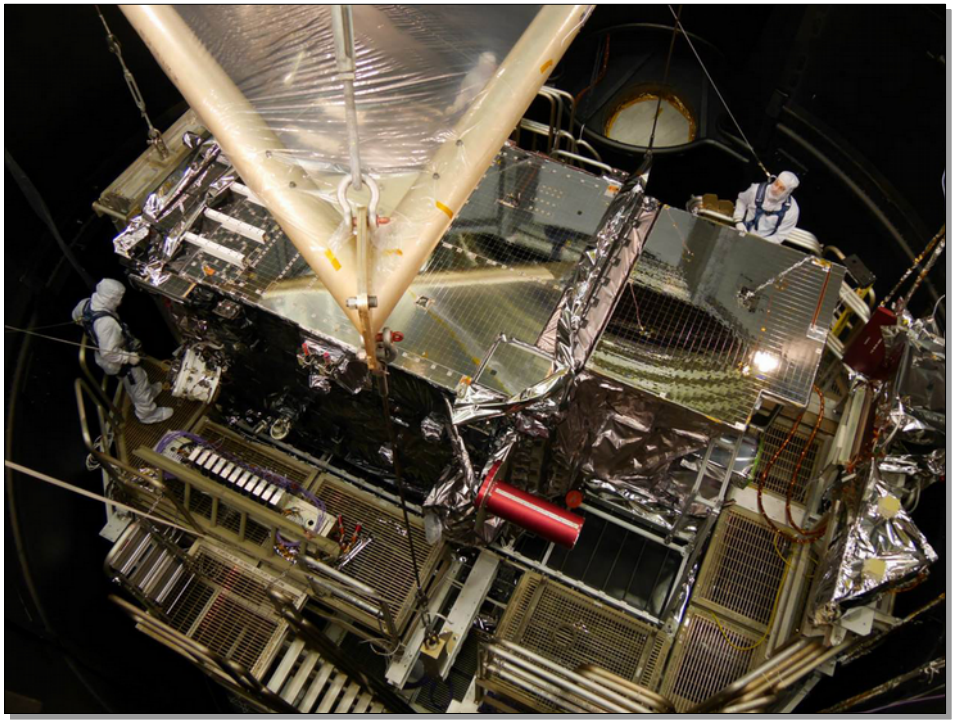
Update on the Contamination Transport Simulation Program (CTSP)

Lubos Brieda
Particle In Cell Consulting LLC
Westlake Village, CA, USA

particleincell.com/ctsp/

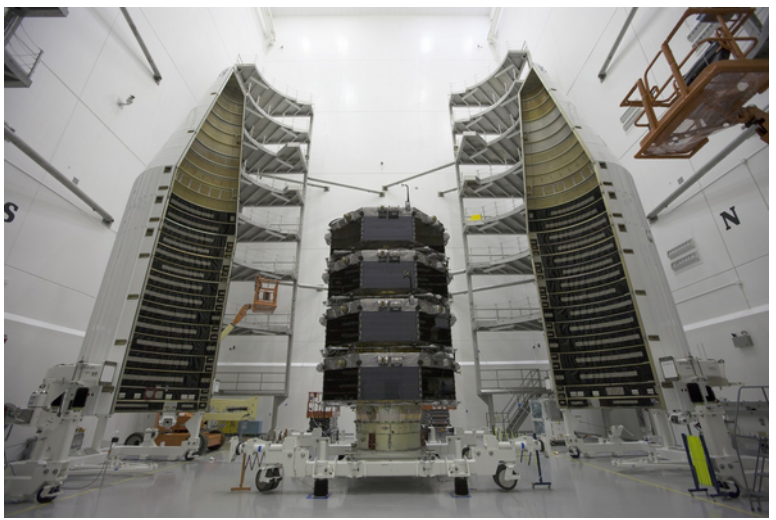
Motivation



Contamination engineers often tasked with predicting contamination transport on complex systems. Complexity arises from:

1) Real-world geometry
2) Transient, multiphysics interactions

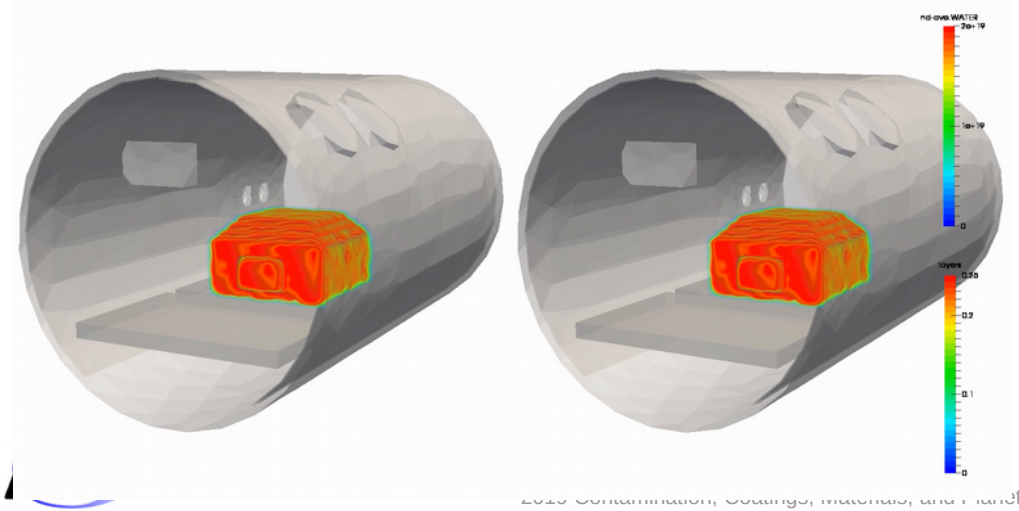
Examples: TVAC chamber full of MGSE producing or reflecting outgassed molecules; particulates released during launch are transported by aerodynamic loads



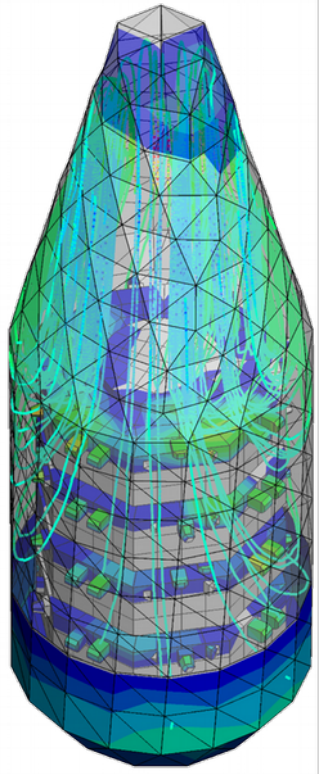
PIC-C developed "CTSP" code attempts to provide the community with a single tool for predicting molecular and particulate transport. These slides present the latest code features.

CTSP Overview

- Originally developed in 2015, presented at the 2015 CCMPP workshop
  - Described in Brieda, L. "Numerical Model for Molecular and Particulate Contamination Transport." J. of Spacecraft and Rockets, V.56, N.2, 2018
- Based on past experience with Particle in Cell (PIC) codes for spacecraft plasma propulsion plume interactions with spacecraft components
  - Brieda, L., Development of the DRACO ES-PIC Code and Fully-Kinetic Simulation of Ion Beam Neutralization, Master's Thesis, Virginia Tech, 2005
- Uses simulation macro-particles to concurrently push molecules, dust particulates and fibers
  - Detailed geometry surface mesh in common formats including Nastran, UNV, STL, OBJ, and TSS
  - Optional volume mesh can be used to visualize contaminant plume pressures
  - Implements models for outgassing, particulate loading, fibers, effusion, analytical plumes, and droplet evaporation
  - Supports dynamic environmental gravitational and aerodynamic forces, and time varying surface temperature
- Customers: NASA, AFRL, various private aerospace firms



Effect of shroud temperature (warm left, cold right) on water vapor density

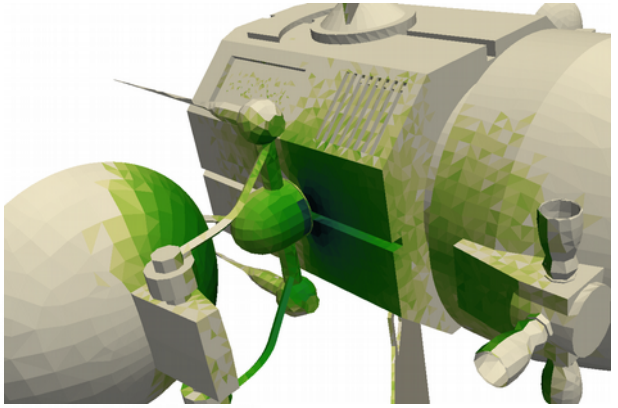


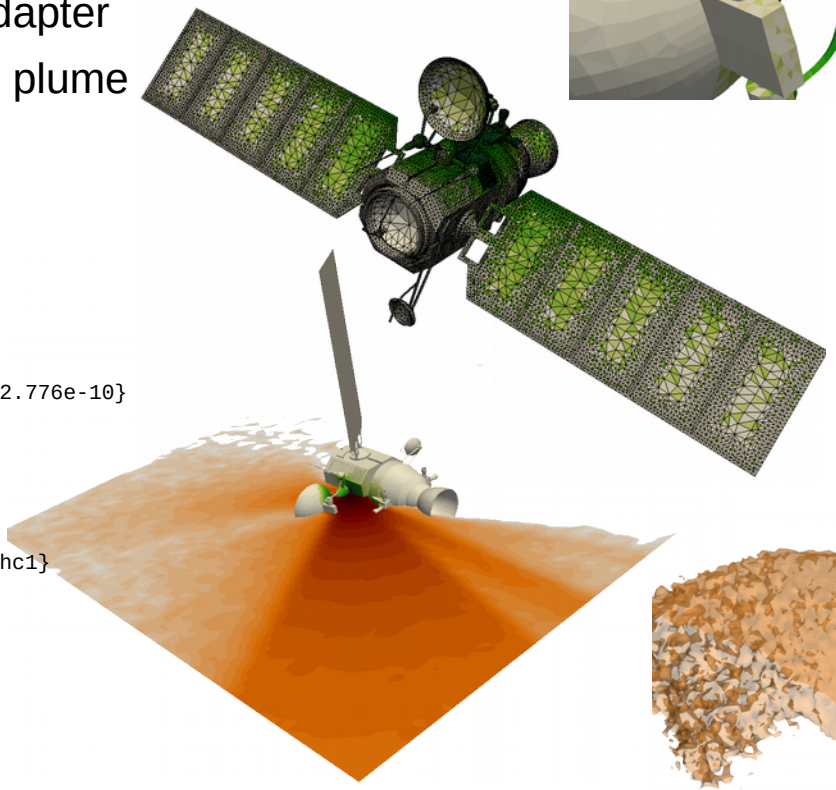
Dust particulate redistribution under ECS flow in launch vehicle fairing

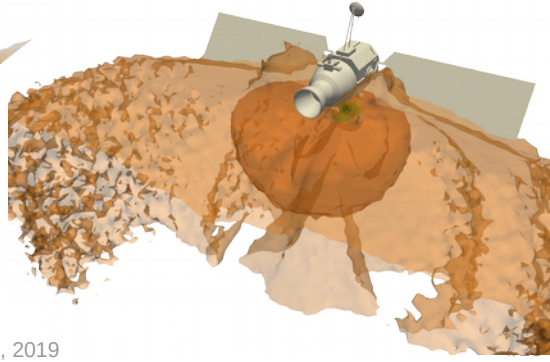
Spacecraft Example

- A "Hello World" example of molecular transport on a generic satellite
- CAD model found online, meshed in Pointwise
- Assigned outgassing source to antenna adapter
- Volumetric mesh used to visualize contam plume

```
#simulation options
options{num_threads:1, file_diag_freq:1}

#load surface mesh
surface_load_unv{file_name:"satellite.unv",view:1}

#define volume mesh for computing contaminant plume
volume_mesh{dx:0.02,dy:0.02,dz:0.02,expand:[0.5,0.5,0.5]}

#specify materials
solid_mat{name:blanketing, weight: 100}
molecular_mat{name:hc1, weight: 54, mpw: 1e8, Ea:12, C_exp:100, r:2.776e-10}

#specify surface properties
surface_props{comps:/.*/, mat:blanketing, temp:260, c_stick:0.3}
surface_props{comps:source, temp:1000, c_stick:0}

#enable outgassing
load_molecules{comps:source, trapped_mass:1e-10, trapped_mats:1.0*hc1}
source_outgassing{model:"exp"}

#run simulation
run_sim{dt:1e-5,num_ts:5000,diag_start:5,diag_skip:5}

#save results
volume_save_vtk{skip:5000,file_name:"field",vars:[nd.hc1]}
surface_save_vtk{skip:5000,file_name:"surf"}
```





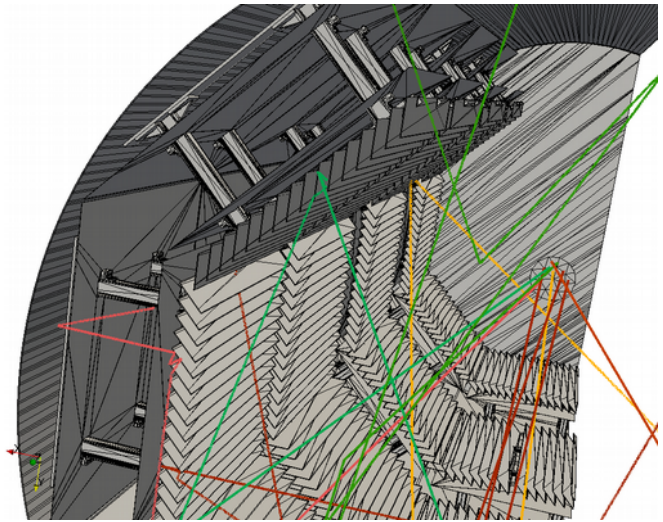


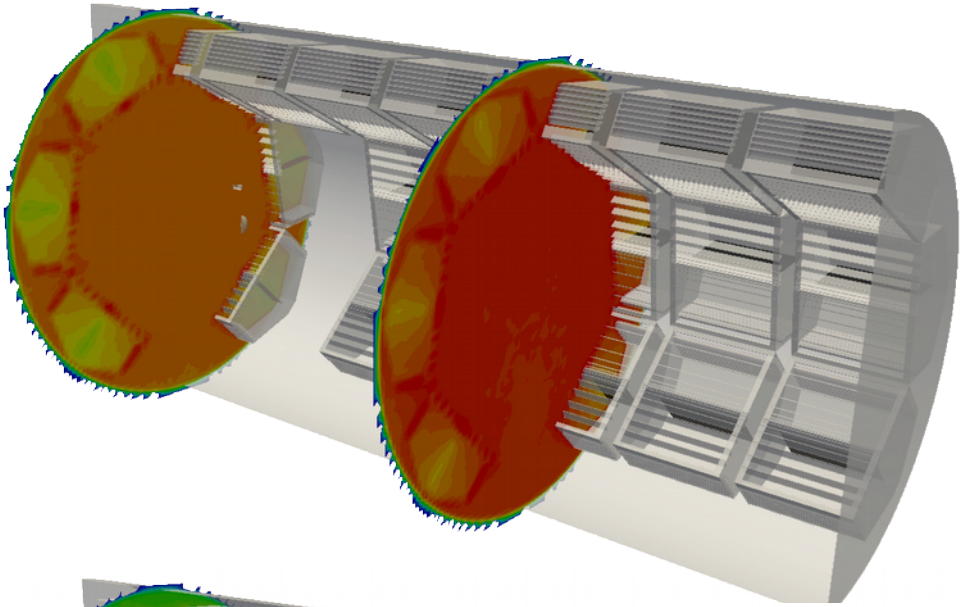
Code Features

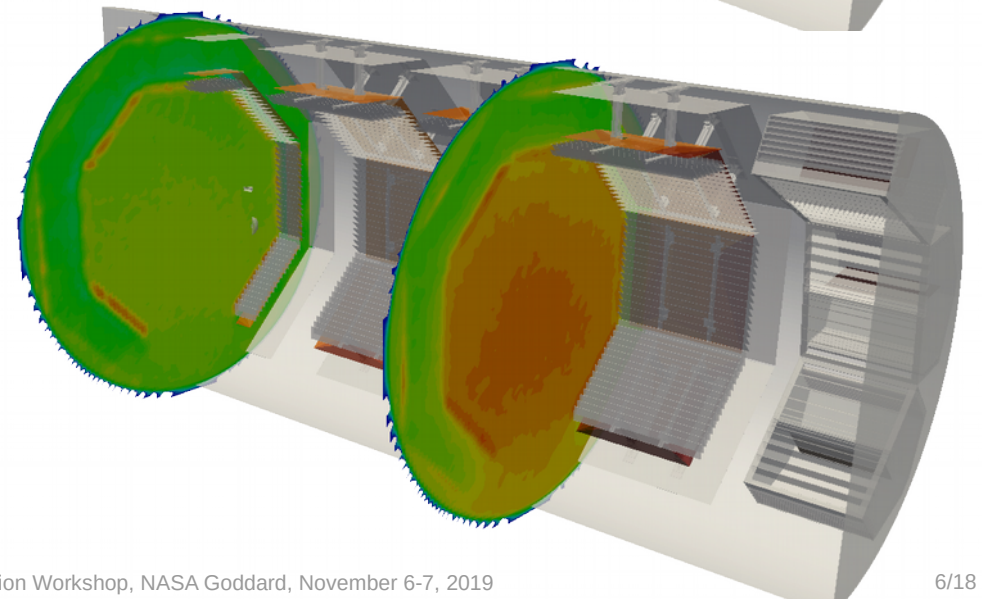
- Three-dimensional simulation code specifically designed for the contamination control community
- Supports complex geometries loaded in common formats such as OBJ, STL, Nastran, UNV, and TSS
- Detailed model for molecular outgassing based on surface desorption and adsorption from the gas phase
- Surface adhesion based on material activation energy and time-dependent surface temperature
- Particulate generation per IEST-STD-1246D, ISO-14644-1, or tapelift data
- Drifting Maxwellian and effusion sources to model venting of internal cavities
- Support for external gravitational, aerodynamic, and electrostatic forces
- All particles traced concurrently, allowing visualization of contaminant plume density, mean velocity, or pressure
- Transition region can be modeled using Direct Simulation Monte Carlo (DSMC) collisions
- Simulation results exported in VTK (ASCII or binary) or Tecplot format
- Available results include time-dependent surface contaminant concentration, histograms of surface particulate population, particle traces, particle scatter plots, and volume data.
- Runs in parallel using multithreading and MPI, written following modern C++11 paradigms

Another Example: Chamber Retrofit

- Aerospace Corp interested in refurbishing vacuum chamber used for electric propulsion testing
- Two options considered: adding more pumps or a custom LHe shroud
- CTSP simulations performed to estimate resulting pressure while EP thruster operated
- Work presented at 2018 Space Prop conference
  - Spektor, R., et.al, “Analytical Pumping Speed Models for Electric Propulsion Vacuum Facilities”,







Numerical Model

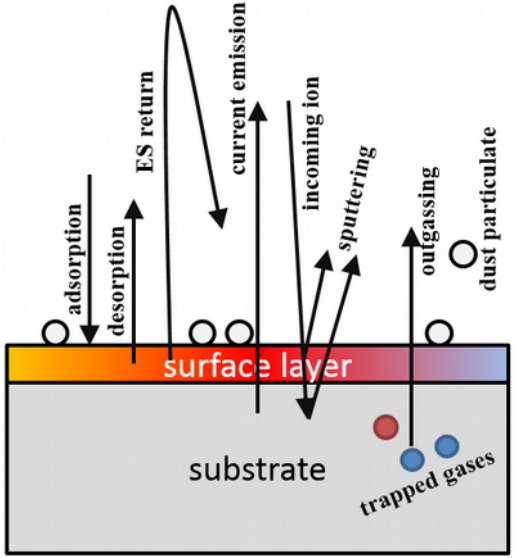
- Molecules and particulates injected into the simulation via surface or volumetric sources
  - Models for outgassing and particulates described on the following two slides
  - Many additional source models available: effusion, droplets, fibers, etc…
- Some sources generate mass continuously (outgassing), while others generate all mass at the start of the simulation (particulate source with default settings)
- All particles advanced through small time steps per equations of motion

$$\frac{d\vec{x}}{dt} = \vec{v} \qquad \frac{d\vec{v}}{dt} = \vec{x}$$

- Velocity updated from force equations
  - Gravity: $\vec{F}_g = m\vec{g}$
  - Electrostatics: $\vec{F}_e = q\vec{E}$
  - Drag: $\vec{F}_d = \frac{1}{2}\rho_g C_d A_d |\vec{v}_a - \vec{v}_p| (\vec{v}_a - \vec{v}_p)$
- Forces evaluated using the specified world or CFD properties
- DSMC used to simulate intermolecular collisions in the transition (Kn~1) regime

Molecular Outgassing / Surface Model

- Objects exposed to vacuum assumed to consist of:
  1. base substrate with mixture of trapped gases
  2. surface layer with mixture of adsorbed materials
- CTSP supports two outgassing models
  - Power law commonly used by the contam community: $\Gamma = C_0 \exp\left(-\frac{E_a}{RT}\right) t^k$ diffusion $k = -0.5$
  - Model of Fang, et.al, $\frac{d\theta}{dt} = -D\left(\frac{\partial \rho}{\partial \hat{n}}\right) + \Gamma_a - \Gamma_d$ with "sorption" function $-D\left(\frac{\partial \rho}{\partial \hat{n}}\right) = h(\rho - \gamma\theta)$
- This mass moved to the surface layer, first-order desorption used to compute number of molecules to inject into gas phase: $\frac{dN}{dt} = \frac{1}{\tau_r} N$
- Residence time computed for impacting particles from surface temperature $\tau_r = \tau_0 \exp\left(-\frac{E_a}{RT}\right)$
- Impacting molecule sticks if $\left(1 - \frac{\Delta t}{\tau_r}\right) \geq R$, otherwise diffusely reflected
  - Residence time thus acts analogously to the "sticking coefficient" used by the community
  - User can also specify a fixed sticking coefficinet to use instead of the temperature based model



Additional processes such as charging, sputtering, or field mession to be implemented in near future

Particulate Model

- CTSP also allows the user to specify initial surface particulate cleanliness level

```
particulate_mat{name:flakes, density:2700}

#load level 600, slope=0.926 particulates, model detachment with Klavins-Lee 4g acceleration
load_particulates_1246{comps:top, mat:flakes, particle_count:200000, level:600, C:0.926}
detach_particulates{model:klavins, klavins_accel_mag: 39.24, release_interval:-1}
```

- This code first generates surface particulates in [1,10), [10,25), [25,50), [50,100), [100,250), [250, 500), [500,750), [750,1000) bins with counts per IEST-STD-1246

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

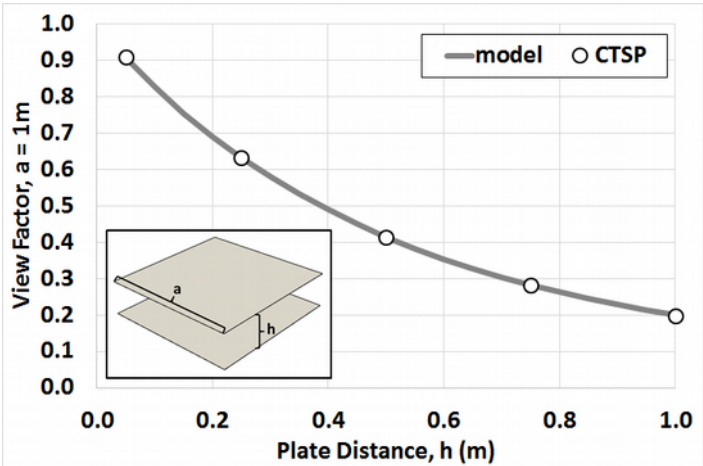
- Model of Klavins and Lee used to compute detachment probability

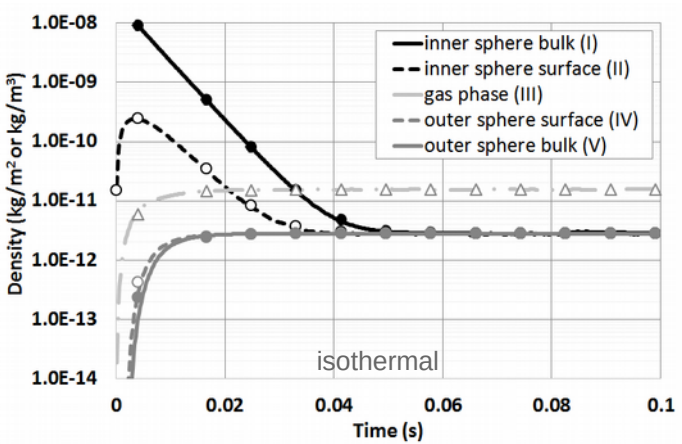
$$\Phi = \left[1 + \left(\left(\frac{a_d}{a_m}\right) / \left(\sqrt{2}\sigma_0\right)\right)\right] /2$$

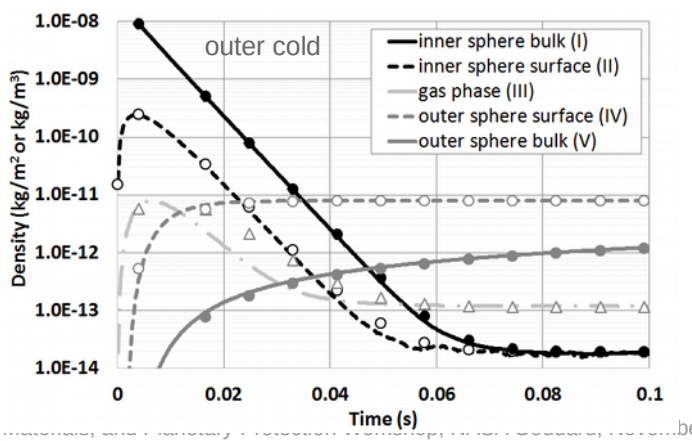
- Particulates modeled as cylinders with spherical caps with aspect ratio per Ma, Fong, and Lee paper
- On surface impact, particulates bounce off with new velocity $v_2 = \alpha_{cr} v_1$, continue bouncing until $v_2 < v_{min}$ (typically 1 mm/s)

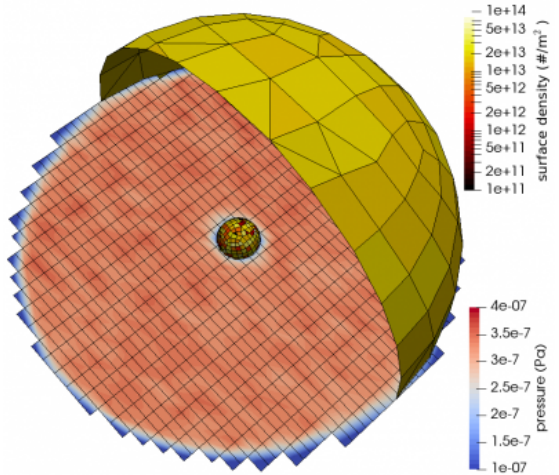
Molecular Transport Test Cases

- Examples discussed in more detail in 2018 JSR paper
- In the first example, the viewfactor between two flat plates is compared to an analytical form, excellent agreement is seen
- The second example demonstrates the code's ability to reach an equilibrium in a closed system
  - A small solid sphere is placed inside a larger hollow sphere. It initially contains some concentration of a molecular contaminant trapped with its bulk region
  - These molecules desorb and redistribute such that uniform density is achieved on the surface and in the substrate of the two spheres.



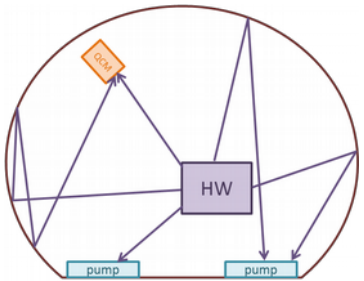




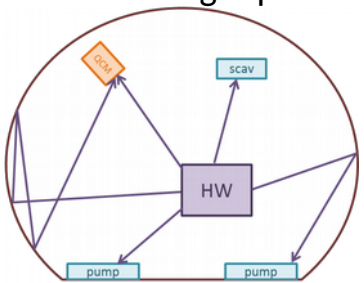


QCM Rate

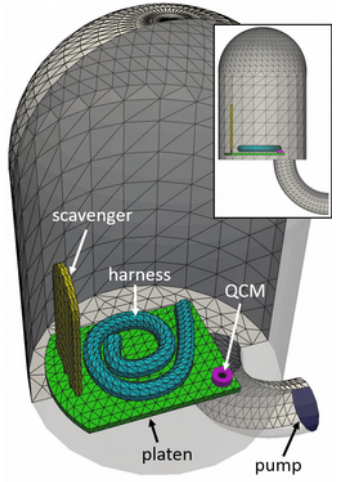
- Another test case demonstrates use of a QCM to estimate outgassing rate
  - In a closed cavity can write a mass balance without and with a scavenger plate



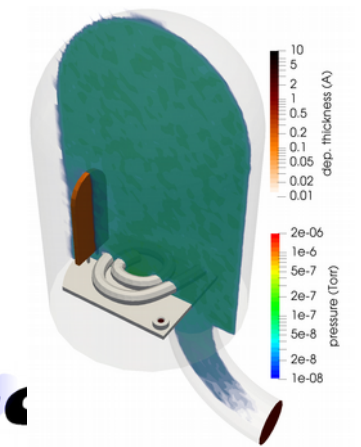
$$\Gamma_h A_h \left(\frac{A_q}{A_q + A_p} \right) = (\Gamma_q)_1 A_q$$

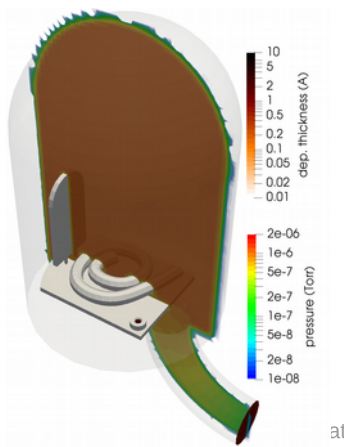


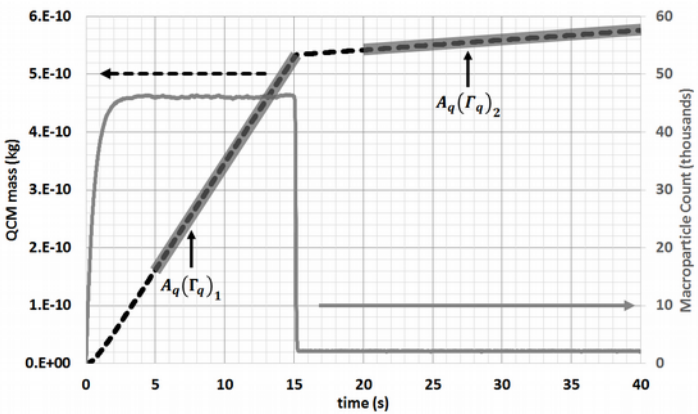
$$\Gamma_h A_h \left(\frac{A_q}{A_q + A_p + A_s} \right) = (\Gamma_q)_2 A_q$$

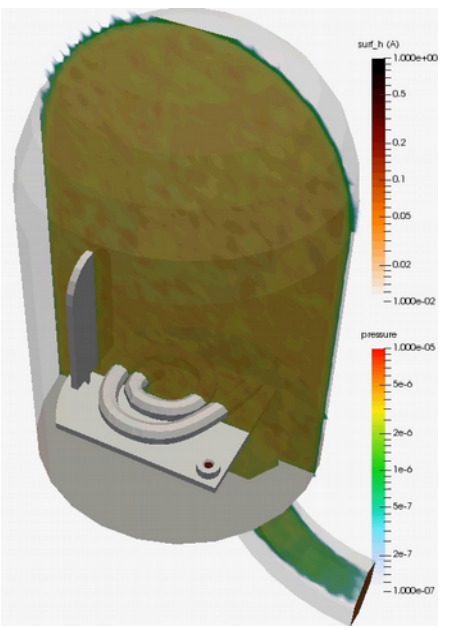


  - Taking advantage of CTSP's time dependent surface temperature to activate the scavenger
  - QCM-computed outgassing rate agrees within 10%, error due to non-uniform scavenger view



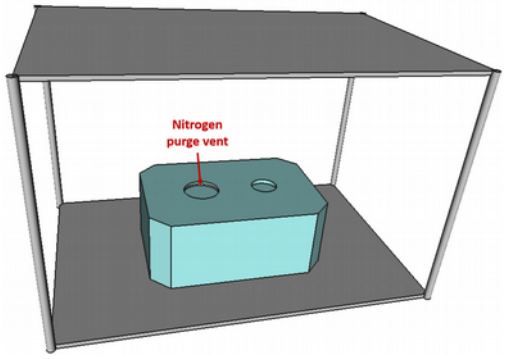


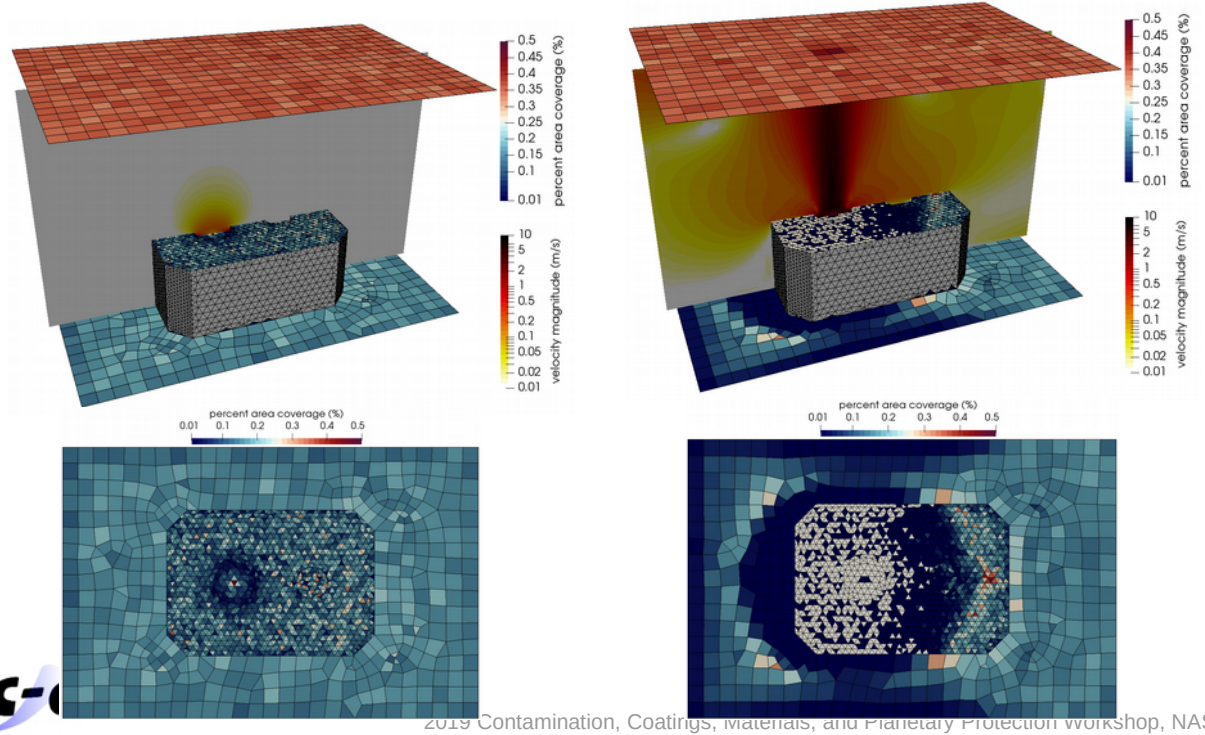


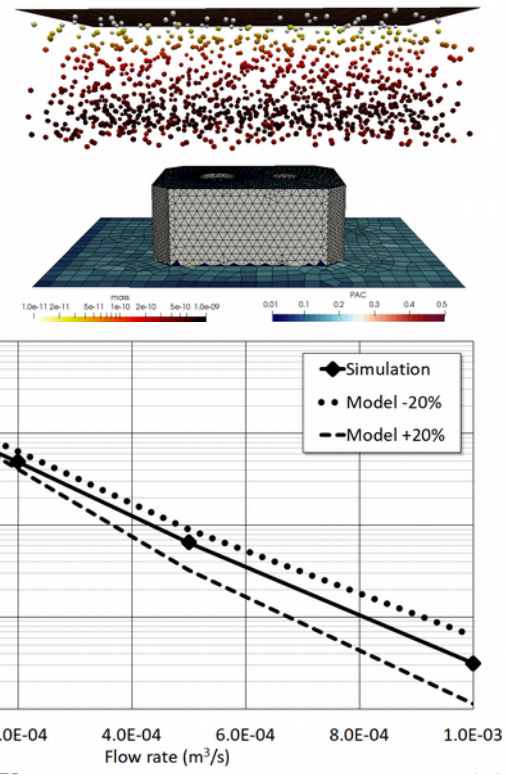


Purge

- Objective of this example is to study the effectiveness of purge on preventing the deposition of particulates onto a detector
- Particulates released from the “top shelf” and we investigate detector cleanliness level using several different purge flow rates
- Results compared against a simplified force balance model

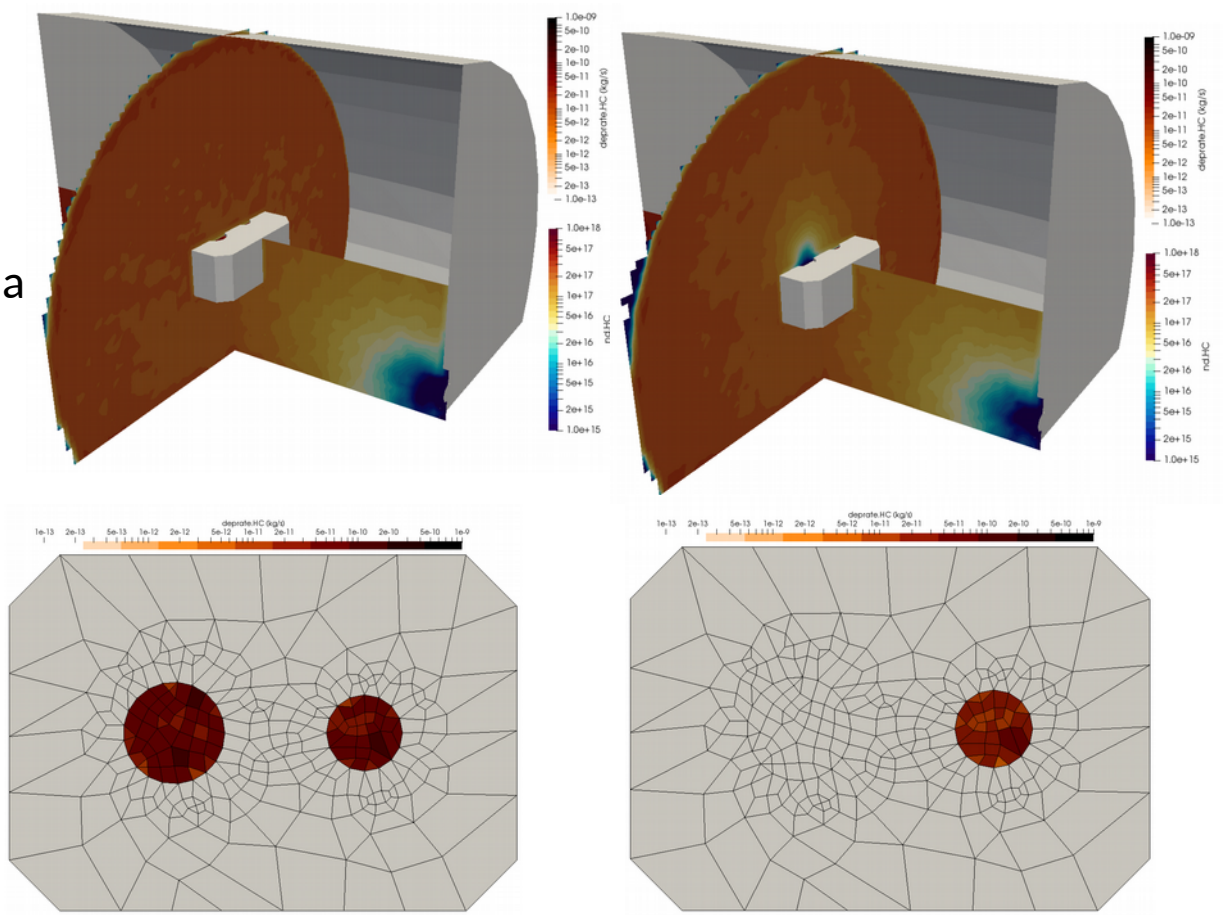






Chamber Repressurization

- This same instrument geometry used in the 2017 CCMPP workshop to illustrate the effect of purge during chamber repressurization
- Chamber assumed to contain some initia concentration of contaminant
- Influx onto the detector during repress reduced with active purge
- DSMC collisions used to capture intermolecular coupling

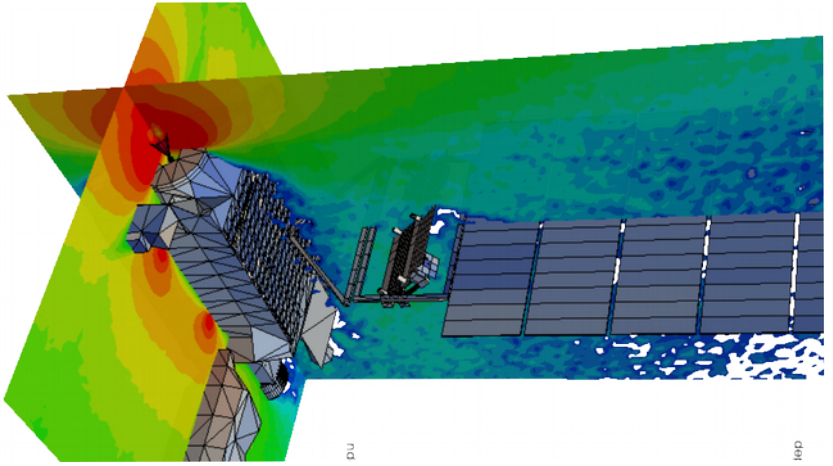


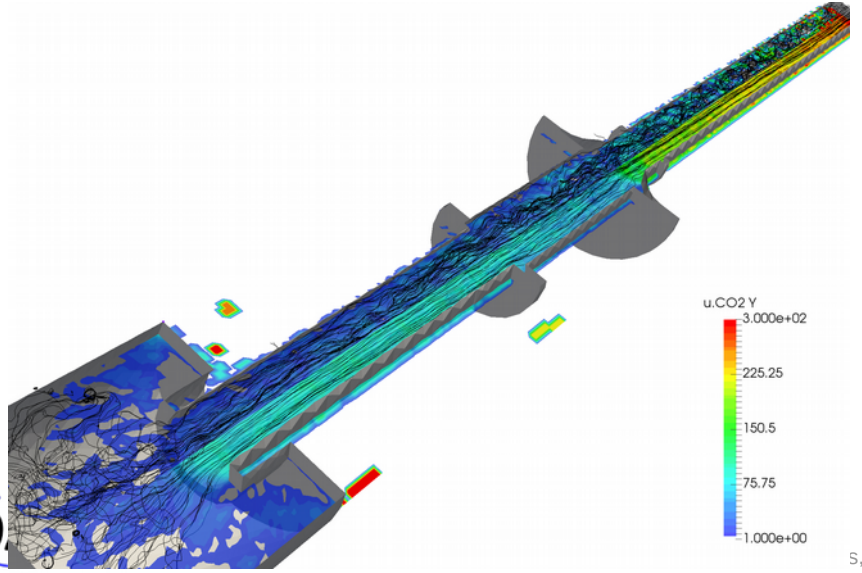
without instrument purge

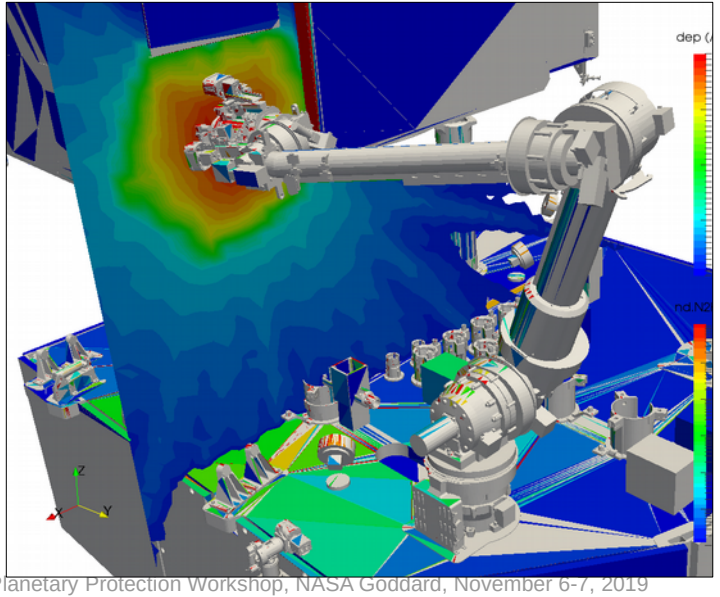
with instrument purge

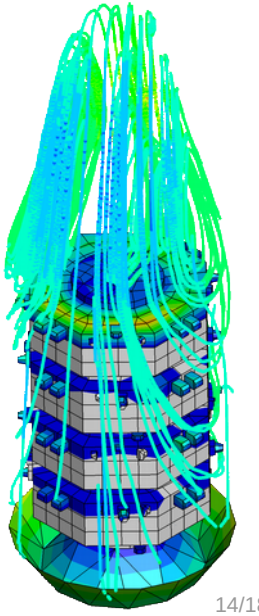
GSFC Examples

- Some selected NASA Goddard applications:
  - Molecular outgassing on GOES-R
  - Particulate transport during launch of MMS
  - Hydrazine evaporation during Restore-L refueling
  - Gas transport in the MOMA mass spectrometer



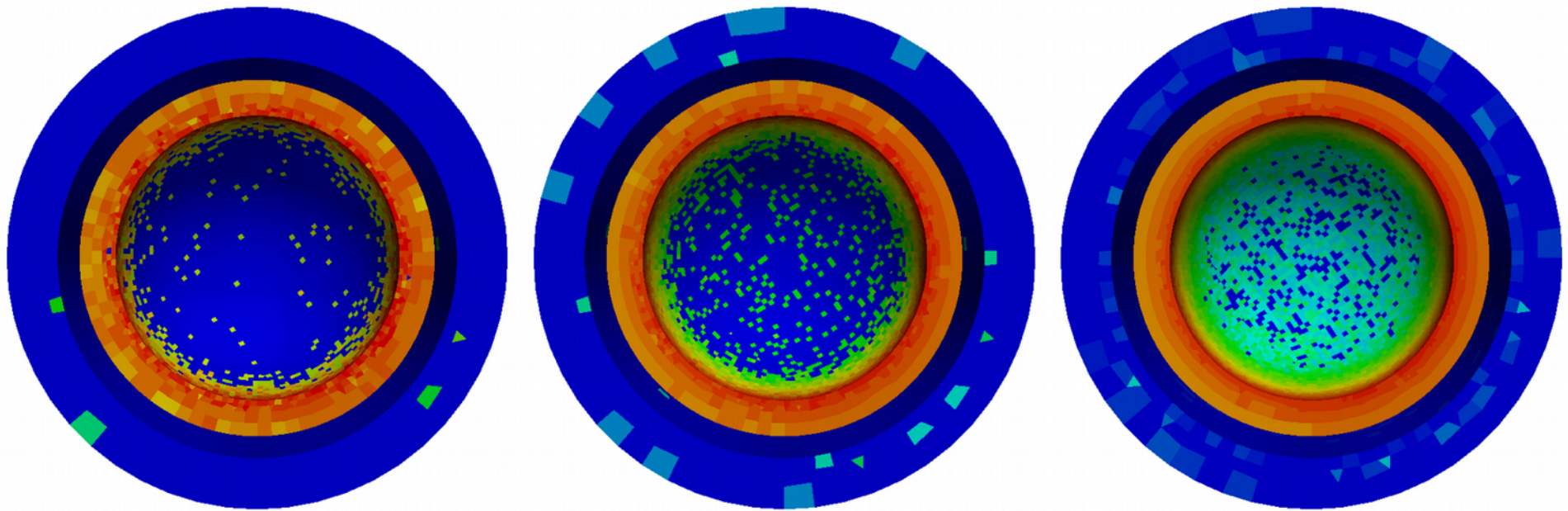






Parallelization

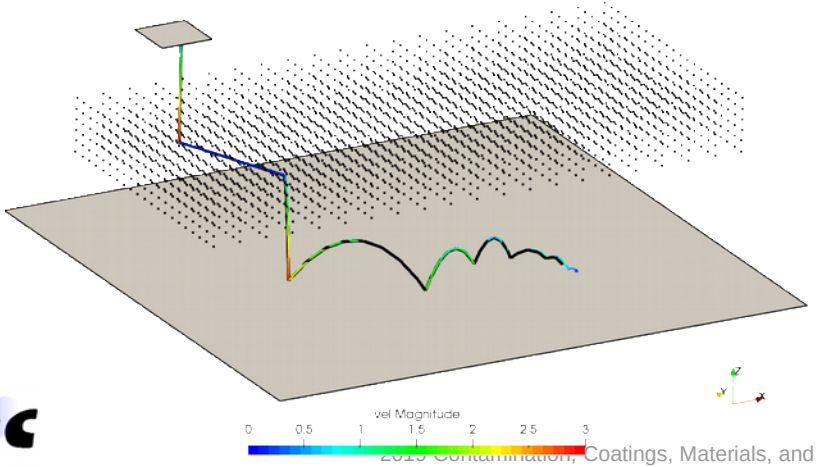
- CTSP supports two types of parallel processing: **multithreading** and **MPI**
- Multithreading takes advantage of the multiple cores found on all modern computers to push particles in parallel
- MPI is used on multi-node clusters. CTSP uses MPI to run multiple "complete" simulations in parallel. The results are then ensemble averaged.

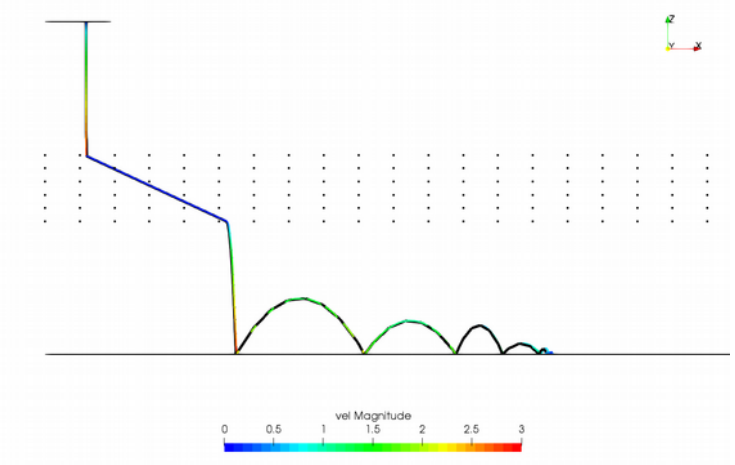


Use of CTSP's MPI ensemble averaging to improve resolution. Log-scale result with 1, 6, and 60 processes.

Subcycling

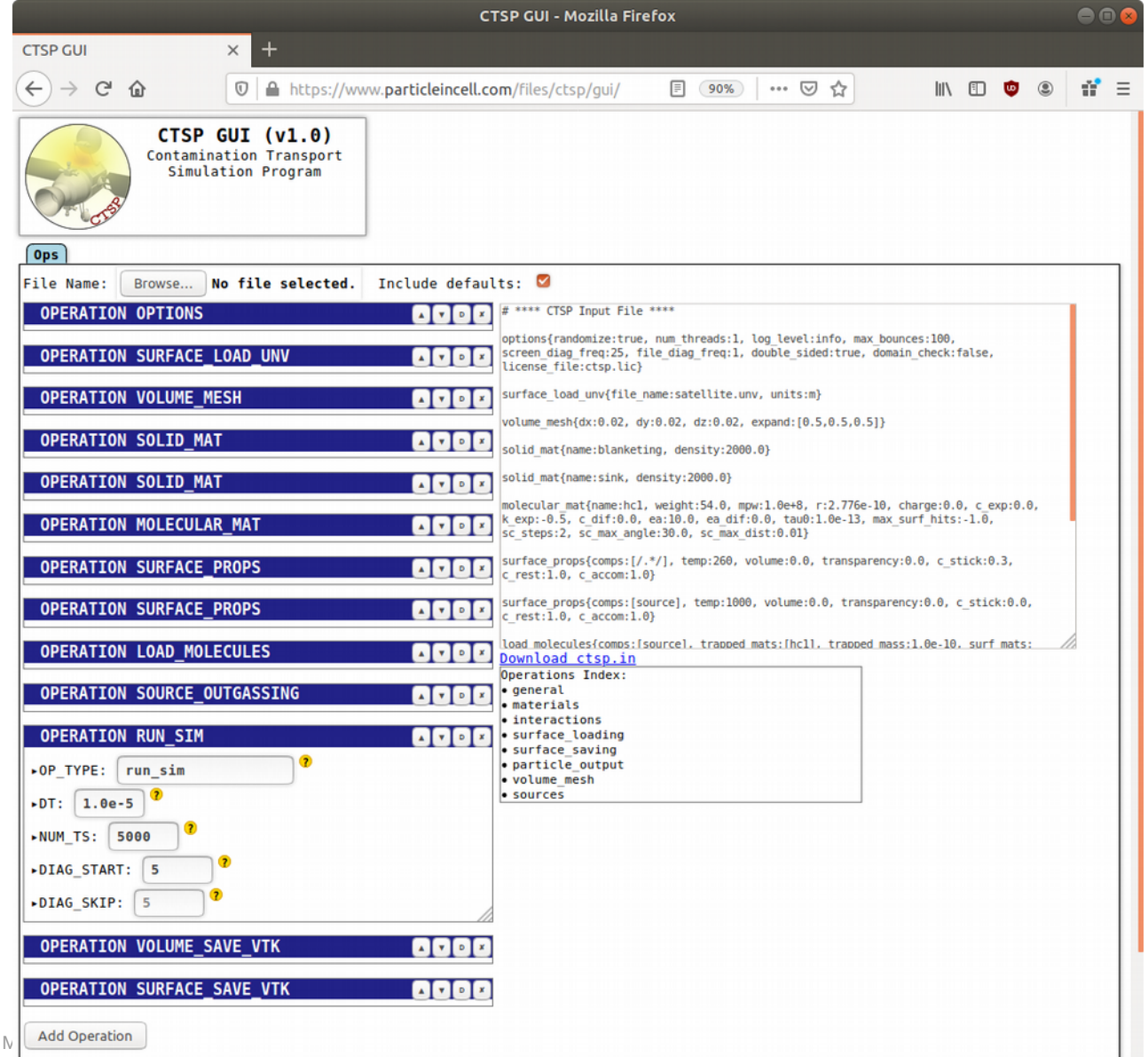
- One past challenge has been the lack of a clear process for setting the simulation time step
- In grid-based codes, can set dt such that particles travel only one cell per time step. Not an option in CTSP since grid-free.
- Issue for particulate sims since small vortexes may be missed if time step too large
- Recently implemented an algorithm to automatically reduce the time step
  - Pushes particle using a single dt, and two half-dt steps. If difference greater than tolerance, the same algorithm is recursively applied to the two half-step pushes.
- Tested by moving particle through an “air pocket”, no difference in large vs. tiny time step





GUI

- A web-based GUI was also implemented recently
- The GUI runs in a browser, either locally, or by accessing the code website
- It simplifies the writing of the input files as all available options and commands are readily available



Future Work

- The CTSP contamination transport code in active development since 2015
  - Latest released version "1.0" (prior versions were 0.xx)
  - Recent changes include complete code base rewrite, addition of parallelization, GUI, and additional output functionality
- Future Work:
  - Additional physics: self-consistent particulate detachment, electrostatic return
  - Exploring additional parallelization via GPUs
  - Surface check octree lookup optimization
  - Code validation – experimental data?
- Open to new collaborations

Contact: lubos.brieda@particleincell.com