Num Tips

This page contains a list of brief HowTo, dealing with specific topics on SPIS numerical solvers.
It is a kind of "Frequently Asked Questions" of Num (contrarily to Controlling NUM from UI, which is more its reference documentation).

As explained in UI documentation, local parameters are based on Material/Electric/Plasma properties that can be edited via GUI, and are then to be assigned locally on your CAD model group per group thanks to UI Group Editor. Global parameters are simply edited in UI Global Parameter Editor.

How to define the environment?

Concerning physics, you can define two Maxwellian distributions of ions, and two of electrons in plasma section. The ones of ions can be drifting (LEO). Note that high energy distributions of electrons are used to define the dose throughout coating, used for radiation induced conductivity. Defining more general environments would request to write an environment model (class) richer than the current BiMaxwellianEnvironment.
Concerning numerics, each population can be modelled through Particle-In-Cell model (dynamical kinetic model) or global Boltzmann distribution (fluid equilibrium distribution).

Back to top.

How to define the data I want for post processing?

Data you want to examine at the end of the simulation must be stored during the simulation.
In the Output section of the global parameters, you can define:

- which type of data you want: 2D maps of local data (potential and currents on spacecraft), 3D maps of local data (potential, densities in plasma), time plots of more global data (potential and currents on electric super nodes)
- how frequently you want them stored

- if you want also their logarithm stored (if you don't know how to plot in log scale in your spot processing tool (in SPIS/UI/Cassandra you need to use a jython script (supplied for positive values only), but is not possible in Paraview))
- if you want to improve the statistics of your data by averaging them between steps and/or at the end of the simulation
- the particle tracking (trajectories), but not yet operational

and you can also modify the verbosity level (level of details of the messages displayed by SPIS-Num (in SPIS-UI standard log and SpisNum.log)

Back to top.

How to turn on environment interactions on SC?

Interactions of spacecraft surfaces with ambient and artificial particles are controlled by global parameters of Surface interactions section.
For each interaction (photo emission, secondary emission from electron or ion impact, radiation induced conductivity and ion erosion), you can typically:

- turn it on with a flag
- supply extra physical parameters, such as secondary temperature, sun direction, etc. (correct default temperatures are supplied)
- specify numerical parameters: time step, numerical speed up, super particle densification

Surface conductivity (on coating surface) and volume conductivity (through coatings) can also be turned on (whereas they are not really interactions).
A few extra local material parameters are also in use (as a material Id or locally defined sun flux), but turning on the interactions locally is net yet implemented, even though the local flags already exist.

Back to top.

How to turn on interactions in gas phase?

Interactions in gas phase are controlled by global parameters of Volume interactions section.
A volume interaction is simulated between two different populations (inPop1 and inPop2), and can produce two populations (outPop1 and outPop2).
These populations must be defined there, including the type of particle.
Incoming populations are pre-existing populations (sources, sourcesâ€¦) while outgoing populations will be created as a result of the interaction.
Interaction cross section can be defined as a constant or energy-dependent.
The only type of interaction implemented for now (SPIS v3.6) is charge exchange (CEX).

Back to top.

How to define a source of particles on my SC?

Particle sources are controlled by global parameters of Particle sources on spacecraft section, and local plasma properties.

Local parameters allow to declare where sources 1 to 4 are active, and specify some extra local parameters (flux, temperatureâ€¦), which may be used or not depending on the source type.
For each of the 4 possible sources, the type of source (a class name) and emitted particles must be specified.
Extra numerical parameters (time step, numerical speed up, super particle densification) can be controlled for each source.

Note that due to the unique local choice of source Id (SourceId in local plasma properties), the emitting surfaces of two different sources cannot overlap. This issue is only related to the number of local parameters currently defined in UI, and could easily be solved in future.

Back to top.

How to control the potential on SC?

You must first initialise the potentials on spacecraft. Then you can let them float or not (i.e. have them fluctuate following charge collection and emission).

Back to top.

How to define and connect electrical super-nodes on my SC?

You first define electrical super nodes, which are usually local grounds (typically a solar array ground / SC bus ground / a probe or thrusters independent possibly biased ground) by giving each of them a different node number.
Then, if you want to let them float, you connect them by capacitors, resistors and electric biases by defining an electric circuit.
See Controlling NUM from UI.html#spacecraft and circuit description for details.

Back to top.

How to let SC potential float?

If you want to let your spacecraft potentials (grounds and surfaces) evolve freely following charge collection (i.e. let it float, and probably reach its so said "floating potential" at steady state), you have to set *electricCircuitIntegrate* parameter to 1. Spacecraft capacitance and SC equivalent circuit will then play an important role. See Controlling NUM from UI.html#spacecraft.

Back to top.

How to initialise SC potential?

SC potential can be initialised at a uniform or non-uniform potential depending on *initPotFlag* and *initPot* global parameters (see Controlling NUM from UI.html#spacecraft). If (and only if) local initialisation is selected, the local parameter *SCDiriFlag* will be used.
Note that due to different localisation/centring of elements the initial potential printed by SPIS-Num may not be exactly the one you requested. The initial values are set on SC elementary surfaces, and the values on nodes common to two groups (on their boundary) follow UI field mapping rules (first groups have higher priority). This consequently induces some diffusion from higher to lower priority group.

Back to top.

How to accelerate a simulation (reduce CPU time)?

Several possibilities:

- instead of performing a PIC-PIC simulation (PIC ions and electrons), go for an hybrid PIC-Boltzmann, if you fulfil the conditions for using Boltzmann distribution for electrons (typically no potential barrier nor positive potential on SC (w.r.t. potential at infinity)). Your time step will then no longer be constrained by the electrons but by the ions (much larger!).
- if you must keep PIC populations of relatively different velocities (ions/electrons, or fast/slow ions) your simulation can remain slow because of the small steps of the fast population and the long total duration needed for slow ones to move. If you are only looking for steady state (or not in some other conditions) try using *numerical times*, i.e. having different different integration durations for different populations. See Simulation Control and all xxxSpeedUp parameters in Controlling NUM from UI.html.
- More classical parameter optimisation can of course also help (a lot):
    - particle numbers; there is
        - a general control by *avPartNbPerCell* parameter in Plasma section,
        - individual controls for sources, *sourceFlagX*, and volume interactions, *volInteract* (when different of 1, they modify the number and weight of emitted superparticles), but none for surface interactions
        - interesting monitoring printed on screen during execution: list sizes
    - time steps (and more generally, for better modelling, do not rely on automatic time step)
    - sometimes Poisson Eq. solver parameters (tolerance in iterative method)…

Back to top.

How to select Poisson Solver type?

A non-linear Poisson solver can be used to solve non-linear Poisson equation, i.e. Poisson eq. including a Boltzmann distribution. Set *linearPoisson* parameter to 1, **and** define one or two population of ambient electrons as a Boltzmann distribution (see plasma section). If you only set *linearPoisson* parameter to 1, but do not define any population of ambient plasma as a Boltzmann distribution (PIC only) linear solver will be used (since Poisson equation has no non-linear term!)

NB: You can on the contrary use a linear solver when using a Boltzmann electron distribution, you will simply lose the advantage of the non-linear solver (stability at Debye length smaller cell).

Back to top.

How to control super particle number?

The number of physical particles is ruled by physics (densities, fluxes, etc.). In a PIC code a smaller amount of particles are tracked, each of them representing many physical particles. Particles tracked in the code are called superparticles, and the number of particles a superparticle represents is called its weight. The physical number of particles (fixed by physics) is obtained as the product of the number of superparticles (undetermined) multiplied by their weight (undetermined), or more precisely as the sum of the weights of superparticles when its not uniform.

So, the product *physNb= superparticNb * superparticWeight* is fixed, but can be decomposed arbitrarily in two factors. The code must thus be told how to decompose *physNb* into *superparticNb * superparticWeight*.

The general rule SPIS obeys is to "try to" have avPartNbPerCell superparticles per cell.
It is easily and directly implemented for initialisation of volume distributions.
Some surface distributions (as e.g. LocalMaxwellSurfDistrib) try to obtain this result by injected the right flux of superparticles. It is based on particle velocities and cell sizes at injection, and can yet be inexact farther from the emission zone.
Other sources of particles (surface and volume interactions, other pure particle sources) have their own "best" way of generating some number of superparticles.
Since it is not possible to determine a priori the user's "best" number of particles, in all cases the user is offered the possibility to densify the superparticles, i.e. multiply their number by a densification factor (larger or smaller than 1) with respect to the default "best" number of the code.
Densification is controlled by the following global parameters:

- *sourceFlagx* parameters for source x on spacecraft
- photoElectronDensification, electronSecondaryDensification, protonSecondaryDensification and erosionProductDensification parameters for the products of the corresponding surface interactions
- volInteract parameters for volume interactions

Whereas no special densification exists for ambient plasma populations, which are simply ruled by the general parameter avPartNbPerCell.

Back to top.