SPIS Numerical core

(or SPIS/NUM, or "the solvers")

Control parameters review

J.-F. Roussel, *ONERA / DESP*

Outline

- How tell the solvers what they must do
- Global control parameters
  - Simulation control
  - Plasma/environment
  - Poisson equation
  - B field
  - Spacecraft
  - Interactions
  - Output control
  - Others
- Local data/parameters

How tell the solvers what they must do

- Through the Java source code:
  - directly in the NUM Java source code:
    - the most powerful way, rather easy in Java thanks to a detailed documentation of the Application Program Interface, the API (in Doc/DocSpisNum/API)
    - training done 3 months ago (6th SPINE meeting in Kiruna)
    - documented in the HowTo pages (in Doc/DocSpisNum/HowTo folder):
      - *Java for NUM*: Java basics
      - *NUM architecture*: code architecture)
      - *NUM integration in framework*: practical file integration)
  - through a script language (python-jython):
    - handling top level objects: spacecraft, plasma, particle distribution, ex:

      ```
      spacecraft.turnOnPhotoEmission(sunFlux);
      ```
    - still to be finalised and documented
- More classically through a user interface (UI / GUI):
  - less powerful, but easier for the user
  - a rather detailed UI developed since last meeting at the demand of the community
  - => this presentation

ONERA

Global parameters

- Where to find them:
  - default values:
    - in python file SpisUI/DefaultValues/defaultGlobalParam.py
    - very similar to a list (a dictionary indeed)
    - very easy => don't hesitate to modify their default value
    - but adding/removing some of them must be taken into account in sources (but programmers can do so!)
  - they can be modified with the GUI:
    - via the menu: solver / set global param
    - unfortunately, their order is scrambled in the GUI dialog (to be modified)
- Details about these global parameters in the documentation (Doc/DocSpisNum/HowTo/Controlling NUM from UI. html)
- We review them here as a guideline to discuss most solvers features

Global parameters: simulation control

Name	Description	Default value	In use
duration	Duration of the simulation	0.1	Yes
plasmaDt	Time step for plasma dynamics (automatic if = 0)	0.0	Yes

- *plasmaDt* = time step for plasma integration:
  - difficult to estimate (plasma may get unstable)
  - hence an automatic assessment is proposed (CFL condition):
    - cross only a fraction of a cell for PIC populations
    - if cell size < Debye length (user responsibility) => dt < 1 / plasma freq.
  - May yet not be very robust

Global parameters: plasma-environment

Name	Description	Default value	In use
environmentType	Name of the Environment class to be used (ex: BiMaxwellianEnvironment which will use the parameters below, or WorstCaseGeoEnvironment which will be self-contained)	BiMaxwellianEnvironment	Yes
electronDensity	electron density (1st population)	1.0e6	Yes
electronTemperature	Electron temperature(1st population)	1.0	Yes
electronDistrib	Name of the VolDistrib class to be used for electrons	GlobalMaxwellBoltzmannVolDistrib	Yes
ionDensity	ion density (1st population)	1.0e6	Yes
ionTemperature	Ion temperature (1st population)	1.0	Yes
IonVx, IonVy, IonVz	Ion drift velocity along x, y, z axes (1st population)	0.0	Yes
ionType	First ion population (a string that must be found in the particle types filename below)	H+	Yes
ionDistrib	Name of the VolDistrib class to be used for ions	PICVolDistrib	Yes

… + 2nd populations

ONERA esa

Global parameters: plasma-environment (cont'd)

- Environment:
  - today only BiMaxwellianEnvironment
  - others possible later: inverted-V population (polar environment), predefined worst case…
- Ions/electrons distributions (electronDistrib, ionDistrib...):
  - GlobalMaxwellBoltzmannVolDistrib and PICVolDistrib
  - when possible prefer GlobalMaxwellBoltzmannVolDistrib:
    - much faster (analytical!)
    - more stable
    - conditions: thermal equilibrium of a "closed system": no attractive potential on SC, no potential barriers, no drift velocity
  - note that these parameters are class names => the selected class is instantiated:
    - Java introspection capabilities
    - very flexible end powerful
    - we are re-building a Java interpreter => not go too far in that direction

Global parameters: Poisson eq. solver

Name	Description	Default value	In use
poissonBCType	0- Dirichlet on SC, Fourier on external boundary using the local fields defined through GUI (see later) 1- Dirichlet on SC, Fourier on external boundary with alpha parameter mimicking a 1/r decay (~vacuum) 2- Dirichlet on SC, Fourier on external boundary with alpha parameter mimicking a $1/r^2$ decay (~pre-sheath) 3- Dirichlet on SC, Fourier on external boundary with alpha parameter mimicking a $1/r^n$ decay, n being next parameter (poissonBCParameter1)	2	Yes
poissonBCParameter1	Parameter that can be used by some BC types (e.g. $1/r^n$ exponent)		Yes
linearPoisson	0- no: use non-linear Poisson solver 1- yes: use linear Poisson solver	0	Yes

- From the UI, boundary conditions are currently limited to:
  - Dirichlet on the spacecraft (fixed potential), the initial potential being defined in the local parameters (see later)
  - Fourier on the external boundary (mixed Dirichlet-Neumann), mostly with parameters defined so as to give an asymptotic behaviour in r^{-n}

Global parameters: Poisson eq. Solver (cont'd)

- Non-linear Poisson solver:

$$-\Delta\phi = e(n_i - n_0\, e^{e\phi/kT}) / \varepsilon_0$$

  - major advantage: stable even for cells larger than Debye length
  - If the selected (*linearPoisson* = 0), the electron distribution(s) are automatically inserted in the non-linear Poisson solver.

- Other parameters to control the maximum iteration number or tolerance of the conjugate gradient Poisson equation solver, rather for specialists

Global parameters: B field

Name	Description	Default value	In use
Bx	x-component of the magnetic field (uniform over the computation box)	0.0	Yes
By	y-component of the magnetic field	0.0	Yes
Bz	z-component of the magnetic field	0.0	Yes

- Uniform magnetic field from UI
- Local magnetic field supported by the solver => possible via the source code, maybe one day via GUI (the local field exists)

Global parameters: the spacecraft

Name	Description	Default value	In use
electricCircuitIntegrate	Flag controlling SC electric circuit integration: - 0: do not integrate (constant initial potentials) - 1: integrate	1	Yes
CSat	Spacecraft absolute capacitance	1.0e-9	Yes
electricCircuitFilename	Name of the file describing extra electric devices between electric (super-)nodes (RLCV)	circuit.txt	No
sourceType	Name of the SurfDistrib class to be used for an artificial source on the spacecraft (ex: LocalMaxwellSurfDistrib, which will use the "source flux", "source temperature" and "source Mach" user-defined local fields, whereas a specific EP model could only use the "source flux" and define internally its velocity distribution)	LocalMaxwellSurfDistrib	No
sourceParticleType	Type of particles (a string that must be found in the particle types)	Xe+	No

- The SC capacitance, *CSat*, may also be considered as a relaxation parameter towards steady state (smaller => faster)

Global parameters: the interactions

Name	Description	Default value	In use
photoEmission	- if 0, no photo-emission - **if 1, photo-emission is turned on** with the sun direction defined below (no shading for now) - **if 3, photo-emission is turned on** with the sun direction defined below **and photo-electron dynamics is modelled (PIC)** - if 5, photo-emission is turned on with a sun flux defined locally (local parameters) - if 7, photo-emission is turned on with a sun flux defined locally (local parameters)and photo-electron dynamics is modelled (PIC) NB: note each bit meaning: bit0=>on, bit1=>local sun flux, bit2=>dynamics of photo-electrons is modelled	0	Yes

- If photo-electron dynamics is not modelled (option 1), photo-emission results in a constant emission current and the potential may go up indefinitely => option 3 may be preferred
- Photo-electron dynamics is modelled through PIC method => very small integration time step (may be automatic) when on (option 3)

Global parameters: the interactions (cont'd)

electronSecondaryEmission	- if 0, no secondary emission under electron impact - if 1, secondary emission under electron impact is turned on - if 2 or 3, secondary emission under electron impact is turned on and secondary electron dynamics is modelled	0	Yes
protonSecondaryEmission	- if 0, no secondary emission under proton impact - if 1, secondary emission under proton impact is turned on	0	No
volumeConductivity	- if 0, no volume conductivity - if 1, volume conductivity is turned on	0	No
inducedConductivity	- if 0, no induced conductivity - if 1, induced conductivity is turned on	0	No
surfaceConductivity	- if 0, no induced conductivity - if 1, induced conductivity is turned on	0	No
sunX, sunY, sunZ	x, y, z-components of sun direction	0., 0., 1.	Yes

- Secondary emission includes backscatterring, cf next presentation

Global parameters: outputs

Name	Description	Default	In use
scPotMonitorStep	time step for spacecraft ground potential monitoring (0.0 => none, -n => n times)	0.0	Yes
scPotMapMonitorStep	time step for spacecraft local potential monitoring (0.0 => none, -n => n times)	0.0	Yes
scCurrentMapMonitorStep	time step for spacecraft local currents monitoring (0.0 => none, -n => n times)	0.0	Yes
plasmaPotMapMonitorStep	: time step for plasma potential monitoring (0.0 => none, -n => n times)	0.0	Yes
densitiesMapsMonitorStep	: time step for densities monitoring (0.0 => none, -n => n times)	0.0	Yes
particleTrajectoriesNb	number of particle trajectories per PIC population	0	No
materialPropertyPlots	plot material properties? 0=no, 1=yes	0	Yes

Global parameters: others

Name	Description	Default value	In use
verbose	Verbosity level (level of screen messages about code execution): 0 = no print at all 1 = prints errors and warnings only 2 = 1 + minimal information 3 = 1 + more information (remains yet readable) 4 = even more information … (next levels for debugging)	3	Yes
poissonVerbose	Same as verbose, but specific to Poisson solver	3	Yes

Local parameters

- Each field is defined on 1 of the 3 meshes: volume, spacecraft, or external boundary
- Each has a given centring/localisation: 0=node, 2=surf, 3=cell
- How to define them:
  - Through the group editor of the GUI => same value over each group
  - Cf previous presentations on the GUI
- Details about these local parameters in the documentation (Doc/DocSpisNum/HowTo/Controlling NUM from UI. html)
- We review them here more briefly than global ones, since they are closer to data than control flags

Most important local parameters

- Material model Id:
  - only 0 today: basic model based on NASCAP properties
  - other values possible when next models available
- Material Id (within the material model): as of today selects one of the NASCAP properties set (19 properties)
- Material thickness (overrides the one in NASCAP properties list)
- Temperature
- Sun flux
- Poisson eq. Boundary Conditions data (potential on SC…)
- Particle source data

…

cf documentation