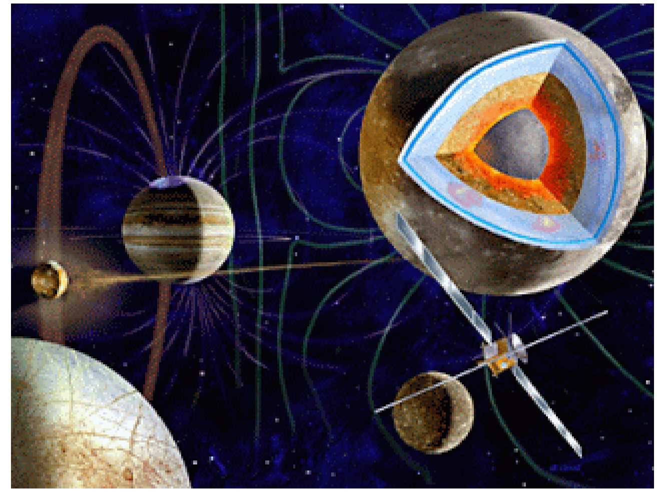
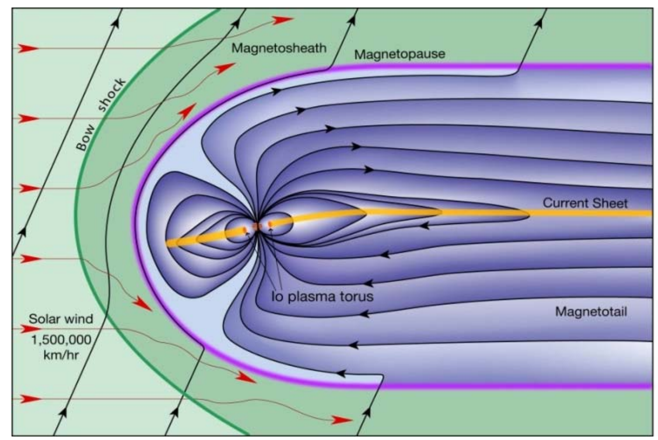
JCAT Project: Monte-Carlo Internal Charging Tool（MCICT） - status report





Fan Lei (RadMod) & David Rodgers (ESA)

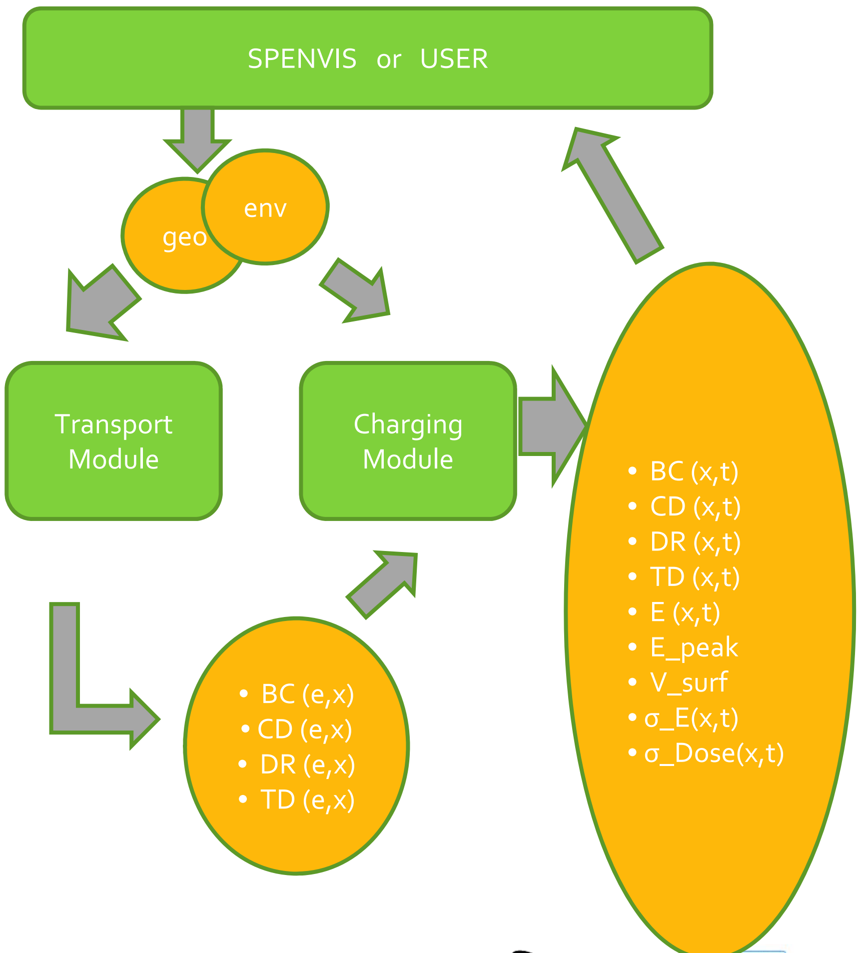
JUICE Charging Analysis Tools (JCAT)

- Involving:
  - Kallisto Consultancy (UK) - PM: Pete Truscott
  - IRF (S)
  - MPS (D)
  - DHC (B)
  - RadMod (UK)
- Aims:
  - Model the charging environment (surface and internal) around Jupiter
  - Create a 1-d internal charging tool to replace DICTAT
    - Requirements – Kallisto
    - Design - RadMod
    - Development – RadMod
    - Testing – Kallisto
    - SPENVIS integration - DHC

MCICT - Introduction

- Existing tool (DICTAT) has weaknesses
  - dose calculation, high-Z materials
- JUICE mission will experience high charging current and dose
- Main objectives
  - New 1D Internal Charging Tool, to be used for the JUICE mission and other environments with high radiation-belt electron fluxes
  - Better particle transport treatment than in DICTAT – MC approach – hence improved current deposition and dose
  - Multiple layers of shielding and dielectrics – more layers and more materials than DICTAT
  - DICTAT charging physics model as baseline, but extendable to improved algorithms or models.
  - Time-dependent environments
  - Performance acceptable compared to DICTAT
  - To be accessible via SPENVIS (https://www.spenvis.oma.be/)

Top level design



- Will be a stand-alone tool and be integrated into SPENVIS
- Separate treatment of particle transport and electric field calculation
  - Geant4-based MC simulation of the incident charged particle transport
  - Ohms-law based electric field solver – similar to that of DICTAT
  - Can be integrated with other particle transport or field-solver modules
- BC – beam current; CD – charge distribution; DR – dose rate; TD – total dose; σ_E – field induced conductivity; σ_Dose – dose induced conductivity ...

Transport Module

- Updated GRAS
  - Slab/Cylindrical geometry – arbitrary layers of shields and dielectrics
  - Per layer based tallies
    - 1 d mesh – n bins:
      - Beam current
      - Charge deposition
      - Dose rate
      - Total dose
- Outputs: Response Functions
  - For BC, CD, DR, TD (E (or constant), x)
  - Energy range: e 100keV – 50 MeV (or limited by the environment inputs)
  - Particle species: e, (p, alpha)

Charging Module

- Inputs: BC, CD, DR, TD(x,t)
  - Convolution of the Response Functions and the Environments
- Field Solvers
  - Re-implemented DICTAT4 model as the baseline solution
  - Other solvers, in future ...
    - New ONERA 3D charging physics model.
    - Ohm's law using finite-difference (finite-volume) method.
    - Poisson's equation and carrier continuity equation , models of carrier mobility and carrier trapping, de-trapping and recombination.

Current Implementations (I)

- Use cases - all driven by the environments
  - Steady spectra - one response function for each particle type (e- and proton)
  - Variable spectra - response function generated at discrete energies
- Python
- User Inputs:
  - Env
    - Particle type
      - constant E and F
      - Const E, F(t)
      - E(t), F(t)
  - Geo
    - layer 1, thickness, material, D/C/O, G/Voltage, n-sub
    - ....
  - Mat - None G4
    - Name, Chemical-Formula, Density, Temperature
  - Dielectrics - new
    - Name - match the rad-mat name
    - 6 -7 other parameters (DICTAT format)
  - Setup
    - Exposure times
    - Decay times
    - Observation times

Current Implementations (II)

- Outputs
  - at each observation time
    - charge distribution in each dielectric layer
    - dose distribution in die-layer
    - E distribution
    - Max E, surface-V in each die-layer
    - charge in floating conductor layer, and voltage
  - For each die-layer, at each observation time
    - E max
    - V surf
  - Each floating Conductor layer

Input file format: Environment

There are 9 options to specify the environment:

```
###Env###
############ format for the env specification ##############
#
#particle, Energy_unit, Flux_units, angle_type(normal/iso)
#Option x
#
#Option 1: Mono, Const
#Mono, Energy, flux
#
#Option 2: PoW, Const
#Pow, alpha, Emin, Emax, f
#
#Option 3, Exp, Const
#Exp, E0 , Emin, Emax, f
#
#Option 4, Arb, Const
#Arb, E1, …, En, F1, …,Fn, flux
#
#Option 5: Mono, Variable
#Mono, Energy
#t1, f1 (time in seconds, f integral omni-directional flux)
#..
#tn, fn
#
#Option 6: PoW, Variable
#Pow, alpha, Emin, Emax
#t1, f1
#..
#tn, fn
#
#Option 7, Exp, Variable
#Exp, E0 , Emin, Emax
#t1, f1
#..
#tn, fn
#
#Option 8, Arb, Variable
#Arb, E1, …, En, F1, …,Fn
#t1, f1
#..
#tn, fn

#Option 9, Table, Variable
#t1,f1, E11, …, E1m, F11, …,F1m
#t2,f2, E21,, …, E2m, F21, …,F2m
#…
#tn,fn, En1, …, Enm, F21, …,Fnm
#
####### End of Format info ##########
```

Input file format: Others

- Geo
  - Shape: Slab/Cylinder
  - layer 1, thickness, material, D/G/F, G/Voltage, n-sub
  - Layer n, ...
- Mat
  - G4 built-in: No need to specify explicitly
  - Others:
    - Name, Chemical-Formula, Density, Temp
- Dielectrics
  - Name – need to match the Mat name
  - 6 -7 parameters (DICTAT format)
- Observation times (in seconds):
  - Step, start, end, step_size / Table , O1, O2, O3,..., On
  - Events, modular, total_events, true/false

Input file example: MM03.inp

```
###Env###
e-, MeV, /cm2/s, normal
#Arb, E1, …, En, F1, …,Fn, flux
Arb, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.

###Geo###
# slab/cylinder
slab
# in the case of slab shape, the first layer id the outer most, while for cylinder shape it is the in
#layer_name, thickness, material, temp, D/G/F/S, n-sub (thickness in mm, temp in K, D: dielectric; G:
L1 , 0.36,  Epoxy , 80,  D , 20
L2 , 0.001,  G4_Al , 80,  G , 1

###Mat###
Epoxy H22-C9-N2 1.39

###Die###
# name, permitivity, dark_conductivity, kp, delta, ea
#G4_MYLAR ,     4.70, 1.48e-18, 9.84e-14, 1.77, 0.42
Epoxy,      1.0, 1.0e-12, 1.0e-14, 2.0, 1.2

###Obs###
#step/table
step, 10, 144010, 3600
events, 10000, 100000, true

```

Kallisto Consultancy RadMod Research DH Consultancy IRF MPS esa

Status

- GRAS updates
  - Mesh tally - done
  - Cylinder shape – done
- Response Function Generator
  - Python code - done
- Charging Module
  - DICTAT-like model
    - Prototype python code completed.
    - Currently under verification and validation ...
  - Other models: in discussion with external collaborator...
- First version expected in June 1015
- Will undergo internal JCAT testing before release.