

An Introduction to the **REST-SIM** Simulation Framework

Fan Lei RadMod Research

(and the REST-SIM project team)









Contents

- ESA Cosmic Version 2015-2025
- The REST-SIM Project
- The REST-SIM Framework
- First Applications
- Future Developments





Cosmic Vision Missions programme

ESA's science missions for 2015-2025:

- Over 150 missions have been proposed
- 4 missions have been selected so far:
 - L class: JUICE (L1,2022)
 - M class: Solar Orbiter (MI, 2017), Euclid (M2, 2020)
 - S class: CHEOPS (\$1,2017)
- Other L class candidates:
 - IXO/ATHENA: Advanced Telescope for High Energy Astrophysics
 - LISA/NGO: New Gravitational wave Observatory
- New M3 studies:
 - EChO (Exoplanet Characterisation Observatory)
 - STE-QUEST (Space-Time Explorer and Quantum Equivalence Principle Space Test)
 - MarcoPolo-R (Asteroid sample return)
 - LOFT (Large Observatory For X-ray Timing)
 - PLATO (PLAnetary Transits and Oscillations of stars)





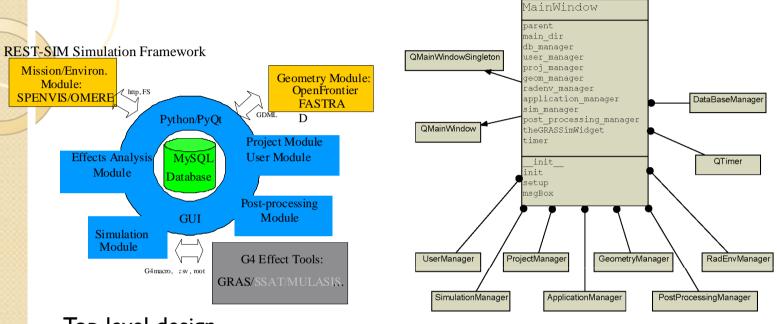


REST-SIM Project

- Radiation Effects on Sensors and Technologies for Cosmic Vision Missions (CVM)
- Main objectives:
 - Survey of the proposed CVM technologies and review their radiation susceptibility
 - Review existing radiation effects analysis tools and capture the SF requirements
 - Design and implement the SF
 - Demonstrate the SF capabilities
- The project team:
 - QinetiQ (now RadMod), etamax, SpaceIT, DHC and UCL/CSR
 - ESA technical officers: Giovanni Santin & Petteri Nieminen
- Main development activities have been completed in 2012

http:/spitfire.estec.esa.int/trac/REST-SIM

REST-SIM Simulation Framework



Top level design

Main class diagram

Key s/w technologies:

- Python and PyQT main programming lang. and GUI
- GRAS/Geant4 particle transport and effects simulation tool
- OpenCascade geometry modelling
- NumPy, SciPy & Matplotlib post-processing
- MySQL internal database

Main tools:







REST-SIM SF Components

- **User manager**: sets up a single user (anonymous, without login) or multi user environment (with registration and login)
- **Project manager**: all configuration and run parameters, user inputs, run results, post-processing products, ... are organised in projects.
- Environment manager: set up connections to a SPENVIS server and import environment spectra from SPENVIS, OMERE or via direct user file import.
- **Geometry manager:** import GDML files or set up, run and import files from ESABASE2 and FASTRAD.
- **Application manager:** set up the parameters for the selected effects tool.
- **Simulation manager:** define host environments for simulation runs and schedule the run execution
- **Post-processing manager:** visualise and plot results, apply response functions and user defined functions.
- **Database:** storage of user defined parameters, imported and generated files, simulation setup and progress, ...







Spacel

REST-SIM SF GUI

Name type application Save input_name Rest Example-1 project Performent env geo Physics Geometry Source Tally Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing analysis Postprocessing Anapped ProstProcessing ProstProcessing ProstProcessing Postprocessing Anapped Anap	REST-SIM					
Name type application Save input_name Reset b Geometry geo b Environment env Simulations Postprocessing analysis pottet.py pytet.py pytet.py pytet.py pottor import numpy as np import numpy as np import numpy as np import numpy as np import pytab as p1 import pytab as p1 import numpy as np import numpy as n	Project Geometry RadiationEnv Application	Simulation PostProcessing				
Example-1 project Geometry geo Environment env Simulations Post-processing analysis Post-processing analysis Post-processin	Project Viewer		₽×	GRAS Input Builder		8 ×
Postprocessing analysis Postprocessing analysis Postprocessing analysis Particle type e. Parti	Name type	at	plication	Save input_r	name	Reset
Post-processing analysis Post-processing analysis Post-processing analysis Post-processing analysis Particle type e- Particle type e- Particle type e- Import pylab as pl import pylab as pl 3 x=2.*np.pi*npl.arange(101)/100 5 pl.plot(x,np.cos(x)) 6 pl.show()	Example-1 proje	ect		Physics Geometry	Source Tally	
Simulations Post-processing analysis Particle type Particle type <td></td> <td></td> <td></td> <td>ingues councey</td> <td>iony iony</td> <td></td>				ingues councey	iony iony	
Post-processing analysis Particle type e. <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
Py_test.py python script Input input Smulation Facility Viewer Smulation Viewer Project Viewer Figure 1 Import numpy as np 2 import numpy as np 2 import numpy as np 2 import numpy as np 3 4 x=2.*np.pi*npl.arange(101)/100 5 p1.plot(x,np.cos(x)) 6 pl.show() Post Processing		wie.		Particle type		
<pre>> Input input Imput input Imput Imput Imp</pre>	dama.			(arouse cype		
Image: Simulation Viewer Project Viewer Simulation Facility Viewer Project Viewer File Editor Image: Simulation Viewer (101) / 100 1 import pylab as pl 3 4 4 x=2.*np.pi*npl.arange(101) / 100 5 pl.plot(x,np.cos(x)) 6 pl.show() Post Processing -0.5						
Simulation Facility Wewer Simulation Viewer Price Editor py_test.py(read only) 1 <td>C2. 192.0374 1. 0.00.000</td> <td></td> <td></td> <td></td> <td></td> <td></td>	C2. 192.0374 1. 0.00.000					
File Editor py_test.py(read only) 1 </td <td></td> <td>ect Viewer</td> <td>M Figure 1</td> <td></td> <td></td> <td></td>		ect Viewer	M Figure 1			
py_test.py(read only) Radius 0.00 * m • 1 import numpy as np 2 import pylab as pl 3 x=2.*np.pi*np.arange(101)/100 5 pl.plot(x,np.cos(x)) 6 pl.show()			A 0 0	+ 🛋 📾 🗸		- m -
1 import numpy as np 2 import pylab as pl 3 x=2.*np.pi*np.arange(101)/100 5 pl.plot(x,np.cos(x)) 6 pl.show() Post Processing -0.5	In case of the second					Radius 0.00
1 Import humpy as np 2 import pylab as pl 3 x=2.*np.pi*np.arange(101)/100 5 pl.plot(x,np.cos(x)) 6 pl.show() 0.0 Post Processing			10			
3 4 x=2.*np.pi*np.arange(101)/100 5 pl.plot(x,np.cos(x)) 6 pl.show() Post Processing -0.5						
5 pl.plot(x,np.cos(x)) 6 pl.show() 0.0 ⊕ Post Processing -0.5				C 3		
6 pl.show() 0.00.50.5		.00	0.5	\backslash	-	
0.00.5 - □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □				X		emax [deg] 0.00
Post Processing -0.5	6 pl.snow()		100000	Δ	/	
			0.0	X	1	pmax [deg] 0.00
	Post Processing		-0.5		-	
Results viewer Python console Plot commands	Results Viewer Python console Plot Commands			1		Response Function mode
				\mathbf{X}		er Law index 1.00
Running 00:05:34	Running 00:0!	5:34	-1.00	1 2 3	4 5 6 7	
225, in <module> 0 1 2 5 4 5 0 7 0 € MeV ▼</module>	225, in <module></module>					po 🔄 MeV 👻
from Ui_SPENVISSetup import Ui_Dialog ImportError: No module named Ui_SPENVISSetup	from Ui_SPENVISSetup import Ui_Dialog			- u		
>>> DataBaseManager.set global current project version id(1)	>>> DataBaseManager.set global current project ve	rsion_id(1)		Load S	nectrum type	SPENVIS trapped electrops
>>> RadEnvDatakeader().fead_ist_of_adenv_mes()	>>> RadEnvDataReader.RadEnvDataReader().read_li	st_of_radenv_files()	*			
Name Type Size Value	Name Type Size	Value	3		environment file	· · · · · · · · · · · · · · · · · · ·
u flashi (101) array([0. , 0.06283185, 0.12566371,						
	0.18849556,	0.25132741,				
			1.12731-2			
			2	N:		

Project manager

Proj	ect Geometry	RadiationEnv Ap	oplication Simulation PostProcessing	
	New	1		
	Open 🔸	wp4200 ►	Version 1 (validation case with)	application
×	Delete		Version 2 (version 2)	
	Export Import			
	Exit			

- Create new and delete existing projects
- Create and delete project versions: creates a copy from a selected base version
- Share/unshare project versions with other users
- Block/unblock project versions
- Export/import projects: write/read all project files into/from a directory tree on disk
- Project viewer: GUI panel listing project version files with rightclick actions: view, edit, delete, run, ...





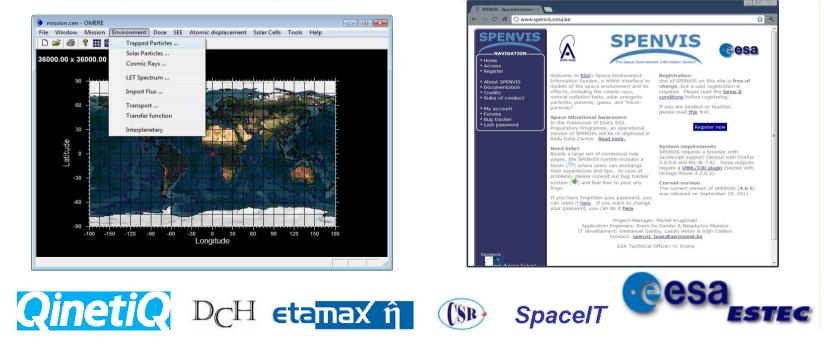
RadEnv Manager

REST-SIM						- • •
<u>User</u> <u>Project</u> <u>G</u> eometry	<u>R</u> adiationEnv	<u>Applica</u>	ation <u>S</u> imulati	on <u>P</u> ostProcessing		
Project Viewer	<u>S</u> PENVIS	•	Import	1		5 ×
Name	OMERE	•	Setup		application	
 example 1 Geometry 	<u>U</u> pload) uet	<u>E</u> dit ►			

Mission environments can be modelled using SPENVIS and OMERE

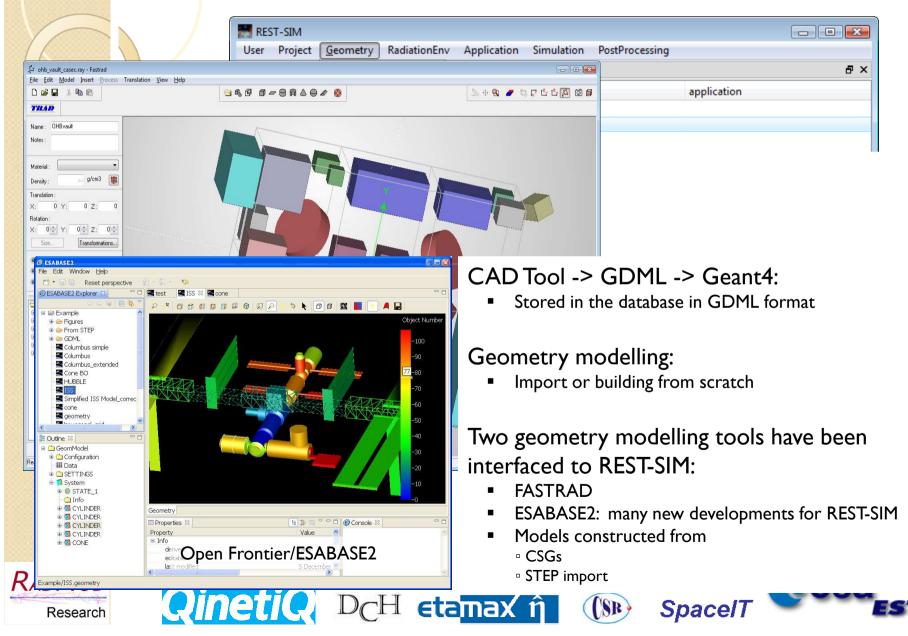
- run from REST-SIM
- environ. data are imported and saved in the project database

User can also upload environ. specifications directly





Geometry Manager



Application/Effects Analysis Manager

REST-SIM				
User Project Geometry RadiationEnv	Application Simulation	<u>P</u> ostProcessing		
Project Viewer	<u>G</u> RAS ►	Import	8 ×	
Name	<u>G</u> RASRMC ►	Edit	application	
example 1	project	Build		
Geometry Environment	geo			
GRAS Input Builder	E	• Ge	eant4 based analysis to	ols:
Save input_name	Reset		• GRAS	
Physics Geometry Source Tally		1	GRASRMC	
Partide type e-]		• (MULASSIS, SSAT, PLA	ANETOCOSMICS)
		• Ge	eometry and Environm	ent from the
Position distribution Point]	dat	tabase	
		• Fu	ll control of Geant4 pł	nysics
Distance from Center 0.00 👘 m 🔻	Radius 0.00 👘 m 🔻		pe of effects/analysis:	
		· //		
Direction distribution mono directional			 Fluence/Current 	Dose
	θmax [deg] 0.00 ♠ φmax [deg] 0.00 ♠		• PHS	
4. [6-2] 0.00 K 4.00 K 0.00 K	11111 (beg]		Dose_equivalent	
	х.		 Equivalent_dose 	LET
	Response Function mode		• NIEL	
E0 0.10 • MeV • Power L Emin 0.01 • Emax 10.00	aw index 1.00		Path Length	
Nb energy bins 10 🔶 🗸 Log Scale	THCY .		= 0	ge_collection
	SPENVIS trapped electrons			-
environment file	· · · · · · · · · · · · · · · · · · ·	Pa	rametric and Templated	d analysis
		1		
Research Research	cH etana	ax n	(SR) SpacelT	esa





	REST-SIM				
	<u>U</u> ser <u>P</u> roject	t <u>G</u> eometry <u>R</u> adiationEnv	Application Simulation	PostProcessing	
Simulatio	Project Viewer		New		₽×
SIIIIUIAUO	Name		type Continu	ue application	
	▲ example 1		project Facility	New	
	Geome	etry	geo Simulat	tion Viewer Management	
Simulation Facility Viewer Name Host name IP Product of the state of the	6] [''] [3000] /home/qinetiq/restsim/	/bin/ /home/ginetiq/	Simulation fac Local ho Linux, or Two executio	cilities: ost, or/and remote (SS r/and Windows (local n modes: ive, forced runs ueue	
Interactive runs			 Load ball 	lance	
Node name	Host name Nr of processors Nr of runs		Results	- auto collection, mer	σρ
Local host local			i courto -		εc
Spitfire spitfir	re.estec.esa.int 16 10	•	Execution mo	onitoring/managem	ent:
			 Check p 	rogress: % completed	d
			•	l/Remove	
			500p/101	Internove	
	Run It Now!				
Simulation V	Viewer				
Name		tatus percentage completed	id 5		
runž	2 1 4572 2011-10-10 16:25:58 ru	unning unning 28.0 % completed in 102.572 :	ls 9		
run] 4 sim-2	h	unning 26.0 % completed in 97.516 s	6		
run runz	2 None None h	nolding None nolding None	10 11		
					ESTEC
Research			Upar Opar		

Post-processing manager

Project Of	eometry F	RadiationEn	v Application	n Simulation	PostProcessing				
Post Processin	g				Import Python Script	1			đΧ
Results View	er Pytho	n console			Plotter				
					Load Results				
Running								Variables 📃 🚌	
Name x Editor	ight", "credit	s" or "license >> >>> [<r Size (101,)</r 	e" for more inform matplotib.lines.Li array([0.	ne2D object at 0		llue 1, 0.18849556,	0.25132741,		
1 imp	ort num		-				*		
3									
4 x=2	.*np.pi	*np.ara	nge <mark>(101) /</mark> 1	00					
	plot(x,	-							
-									
6 nl	Show()								
6 pl.	show()								
6 pl.	snow()								
6 pl.	snow()	r							
6 pl.	snow()	W Fig		•					<u>.</u>
6 pl.	snow()			+ 🥑 (ð 🗸 🖷				3
6 pl.	snow()		00		\$ 4 8				3
6 pl.	snow()			++ @	§ √ 8	, ,			3
6 pl.	snow()		00	+ 💌 ।					
6 pl.	snow()		1.0	+	ĝ ≪ 8				
6 pl.	snow()		00	+	ĝ ≪ 8				
6 pl.	snow()		1.0	+	ĝ ≪ ⊟				
6 pl.	snow()		0.5	+					
6 pl.	snow()		1.0	+ 2					
6 pl.	snow()		0.5	+ @		/			
6 pl.	snow()		0.5 - 0.0 -	+ .					
6 pl.	snow()		0.5	+					
6 pl.	snow()		0.5 - 0.0 -	++ @					
6 pl.	snow ()		1.0 0.5 -0.5	+					
6 pl.	3now ()		0.5 - 0.0 -	+ e i		4	5 6		
6 pl.	snow()		1.0 0.5 -0.5			4	5 6		
			1.0 0.5 -0.5			4	5 6		
			1.0 0.5 -0.5 -1.0 0	i	2 3	•			
6 pl.			1.0 0.5 -0.5 -1.0 0	i		•			

Interactive Python scripts

- NumPy, SciPy, Matplotlib
- Python console and editor

Plotting:

- Id/2d histograms

Post-processing:

- Operation on histograms

Spacel7

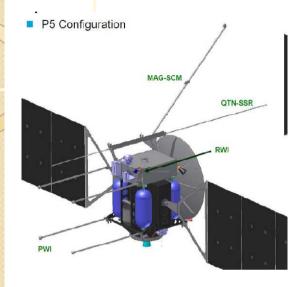
- Derivative parameter analysis
- Analysis based on response functions

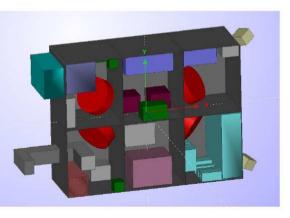
·eesa

- ...

CSR

Demonstration Application: JUICE





- Two demonstration applications:
 - JUICE
 - Solar Orbiter
- JUICE:
 - **Environments: ESA specifications**
 - Geometry Model:
 - Simplified OHB study configuration
 - Detailed geometry model of the • StarTracker/APS
 - Modelled with FASTRAD
 - Analysis:
 - TID the APS/StarTracker, and others
 - Comparison with SSAT results











JUICE - Environments

- ESA Specification:
 - Memorandum on 'Radiation Environment Specification for Jupiter Mission Reformulation Activities' (SRE-PA/2011.050/CE issue 1.3, 10/08/2011)
- Two mission scenarios:
 - Baseline: Callisto + Ganymede
 - Flybys: No Callisto + 2 Europa flybys
- Total proton and electron fluence spectra
 - proton_basline.txt
 - proton_flyby.txt
 - electron_basline.txt
 - electron_flyby.txt

- proton_flyby.txt
- # JUICE flyby proton fluence

(SR)

- # Energy spectrum
- # Energy Differential_flux Integrated_flux
- # (MeV) (MeV-1.cm-2.sr-1) (cm-2.sr-1)
- 0.1 1.17E+14 2.48514E+13 479F+13 1.56082E+13 02 400 5.45E+04 7812679.029 500 2.34E+04 4144082.801 700 6.61E+03 1751316.074 1000 1.57E+03 716314.2436

<u>P</u> roject <u>G</u> eometry	RadiationEnv	<u>Application</u>	<u>S</u> imulation	PostProcessing	
Project Viewer	<u>S</u> PENVIS	+			
Name	<u>O</u> MERE	 type 	2		application
 ✓ wp4200a ✓ Geometry 	<u>U</u> pload		roton spectra		
 ohb_vault_caseg.gdml of_aps_array.gdml FCA_Hydra.gdml 		el	ectron spectra		
		geor	metry file		
		geor	metry file		
aps-array.c	dml	geo	geometry file		
▷ juice.gdml	•	-	metry file		
A Environment		env	,		
UPLOADED	PROTONS	prot			
proton	_flyby.txt	envi	ronment file		
proton	_baseline.txt	envi	ronment file		
	ELECTRONS	elec	tron spectra		
electro	n_baseline.txt		ronment file		
	n_flyby.txt	envi	ronment file		

Spacel

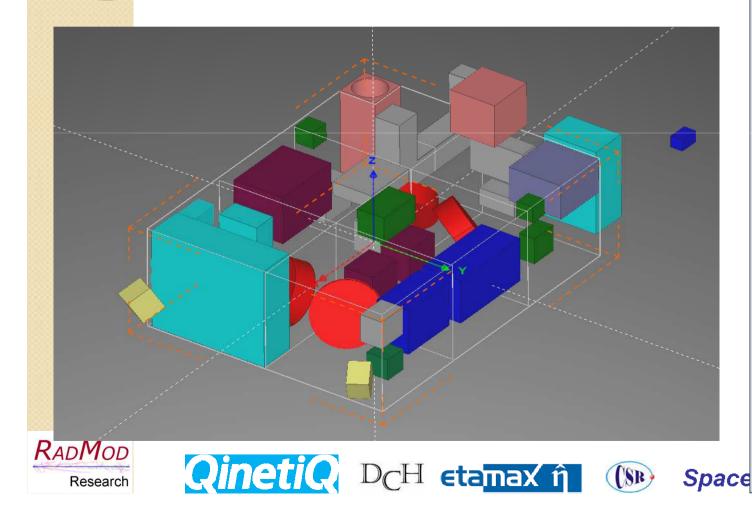
eesa





JUICE Geometry: QinetiQ/OHE

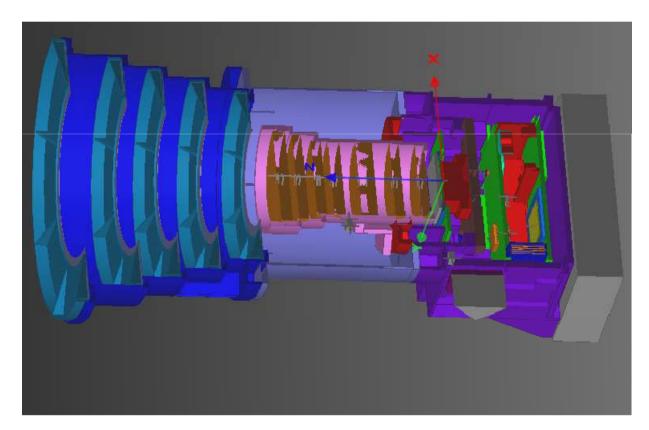
- A FASTRAD model of the service and payload modules of JUICE have been created using FASTRAD, based on the results of a separate study by OHB/QINETIQ
 - Most components are housed in a vault which is not shown



Components List	
🔄 OHBvault	
⊡ © a payload	
⊡ ∰ USO_geom	
⊕ ⊕ PWRI_geom	
In geom	
In the state of t	
⊡ ⊡ CameraDPU_geom	
Image: Constraint of the second state of the second sta	
🖃 🗐 SWIwall 1_geom	
B G SWIwall2_geom	
⊕ [9] SWIwall3_geom	
⊞ ∰ MAG1_geom	
iand Game MAG2_geom	
⊡ ⊡ geom	
⊞⊡grco_geom ⊞⊡g/Battery_geom	
EPC1_geom	
B (B EPC2_geom	
 PCU_geom Battery_geom Soandwaveguide_geom PEPC1_geom PEPC2_geom PEPC2_geom PEPC1_geom PEPC2_geom 	
🗄 🗐 Startracker1_geom	
🗄 🗐 payload_intmass	
🗄 🗗 reaction wheels	
L	

JUICE Geometry: Detailed model of the Star Tracker

A FASTRAD model of the Star Trackers has been developed by SODERN and made available to the REST-SIM project.

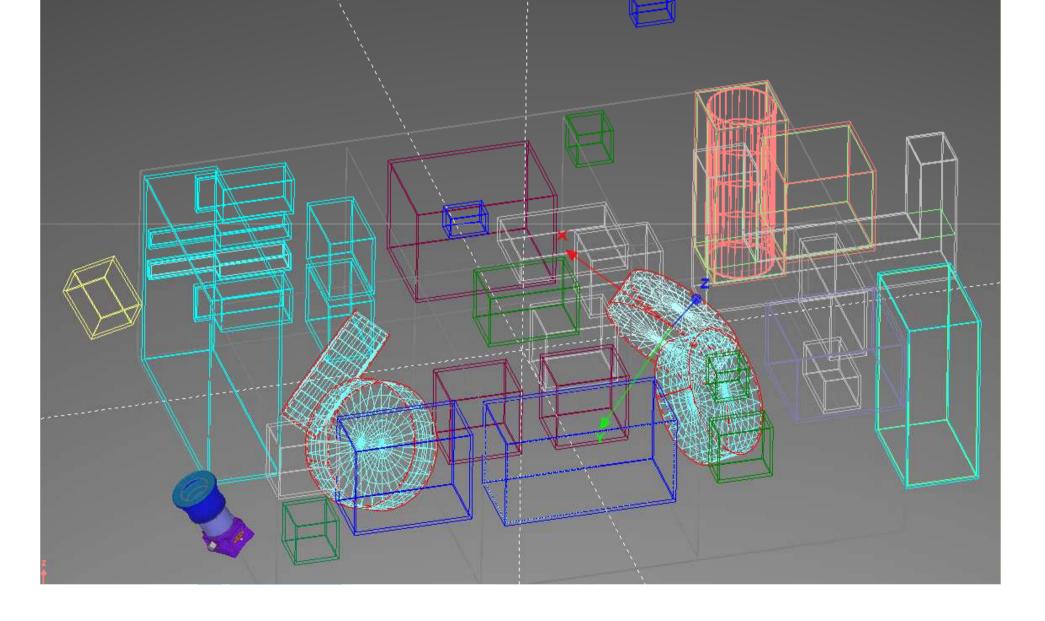








JUICE Geometry: QinetiQ/OHB + SODERN model



TID Analysis with GRASRMC

TID analysis:

ė

- GRASRMC
- QinetiQ/OHB geometry model
- Both baseline and flyby environments
- For 10 different units in the payload and service modules

GR/	AS Reverse Inp	out Builder			
	Save	mag1			Reset
G	GRAS Revers	e Input Builder			
	Sa	ave mag1			Reset
	Geometry	Adjoint/External Source	Primary Spectra	Cuts	Tally

Adjoint Source or Doint Detector

		Project Viewer				
	Dietferm/Deerdeed	Name	type	application	date	id
	Platform/Poayload PCU TWT #1 SMU SG ACS STS AA STR = SG ACS STS AA STR = UVIS2 VIRHIS OU (outside vau	 wp4200a Geometry Environment Simulations Post-processing Input twt.nml 	type project geo env analysis input namelist file namelist file	GRASRMC GRASRMC	date 2012-04-28 19:36:16 2012-04-28 19:40:57	13 2016
	SWI (outside vault) PP DU (outside vault) MAG #1 (outside vault)	smu.nml swi.nml stk2.nml mag1.nml ppdu.nml uvis2.nml	namelist file namelist file namelist file namelist file namelist file namelist file	GRASRMC GRASRMC GRASRMC GRASRMC GRASRMC GRASRMC	2012-04-28 19:40:57 2012-04-28 18:56:40 2012-04-28 19:50:32 2012-04-28 19:03:41 2012-04-28 19:52:52 2012-04-28 19:51:32	2018 1989 2022 1991 2024 2023
		virhis.nml hydra_d1.nml aps-array.nml a-a-e1.nml of_array.nml	namelist file namelist file namelist file namelist file namelist file	GRASRMC GRASRMC GRASRMC GRAS GRAS	2012-04-28 18:34:35 2012-05-04 11:17:28 2012-04-29 00:39:25 2012-04-29 00:48:47 2012-04-29 22:37:54	1965 2158 2128 2140 2150
(Research	pcu.nml stk1.nml	namelist file namelist file III	GRASRMC GRASRMC	2012-04-28 19:35:08 2012-04-28 19:50:22	2014 2021

Simulation executions

	📰 Define a sir	nulation			? ×
	Simula	nulation tion Name: twt XML file: twt.nn	nl (GRASRMC)		-
Project Viewer	Project Viewer				
Name	Name	type	application	date	*
⊿ wp4200a	⊿ wp4200a	project			
Geometry	Geometry	geo			
Environme	Environment	env			
 Simulation 	 Simulations 				
⊳ ppdu	⊿ ppdu	simulation	GRASRMC	2012-04-28 21:03:07	uns
⊳ swi	ppdu1335643387_1212287075_Spectrum3.csv	csv file	GRASRMC	2012-04-28 21:25:07	-
⊳ stk1	ohb_vault_caseg.gdml	gdml file	GRASRMC	2012-04-28 21:25:07	Ε
virhis	ohb_vault_caseg_materials.xml	xml file	GRASRMC	2012-04-28 21:25:07	
▷ uvis2	ppdu1335643387_1212287075.csv	csv file	GRASRMC	2012-04-28 21:25:07	
⊳ stk2	ppdu1335643387_1212287075Convergence_TID.t	xt txt file	GRASRMC	2012-04-28 21:25:07	
▷ mag1	ppdu1335643387_1212287075_Spectrum1.csv	csv file	GRASRMC	2012-04-28 21:25:07	
⊳ twt	ppdu1335643387_1212287075_Spectrum2.csv	csv file	GRASRMC	2012-04-28 21:25:07	
⊳ pcu	ppdu.nml	namelist file	GRASRMC	2012-04-28 19:52:52	
⊳ aa	ppdu1335643387_1212287075_Spectrum4.csv	csv file	GRASRMC	2012-04-28 21:25:07	
▷ of_a	ppdu_1335643387_1212287075.g4mac	g4mac file	GRASRMC	2012-04-28 21:25:07	
▷ hydra_	ppdu_1335643387_1212287075.g4mac.log	log file	GRASRMC	2012-04-28 21:25:07	
⊳ smu	⊳ swi	simulation	GRASRMC	2012-04-28 18:57:16	
Post-proce	▷ stk1	simulation	GRASRMC	2012-04-28 19:55:15	
▷ Input	▷ virhis	simulation	GRASRMC	2012-04-28 18:36:25	
	▷ uvis2	simulation	GRASRMC	2012-04-28 21:02:33	-
•	< III				F I









GRASRMC Simulation Results

			baseline			flyby	
		electron	proton	total	electron	proton	total
	PCU	3.03±0.01	67.1±3.1	70.1±3.1	3.04±0.01	81.1±4.1	84.1±4.1
η.	TWT #1	3.07±0.01	71.1±3.1	74.1±3.1	3.12±0.01	86.1±3.1	89.1±3.1
Platform	SMU	3.01±0.01	63.1±2.1	66.1±2.1	3.05±0.01	78.1±1.1	81.1±1.1
	Star Tracker 1	3.82±0.02	88.1±2.1	92.1±2.1	3.81±0.02	107.1±2.	111.1±2.1
	Star Tracker 2	3.78±0.02	90.1±2.1	93.1±2.1	3.83±0.02	109.1±2.	112.1±2.1
	UVIS2	3.17±0.04	62.1±1.1	65.1±1.1	3.2±0.01	76.1±2.1	80.1±2.1
_	VIRHIS OU	3.38±0.02	124.1±4.1	127.1±4.1	3.42±0.02	142.1±4.	145.1±4.1
Payload	SWI	3.62±0.02	129.1±13.	133.1±13	3.66±0.02	153.1±15	156.1±15.
	PP DU	2.15±0.01	43.1±2.1	45.1±2.1	2.18±0.01	53.1±3.1	55.1±3.1
	MAG #1	4.23±0.03	67.1±1.1	71.1±1.1	4.3±0.03	84.1±1.1	88.1±1.1

TID in krad(Si)

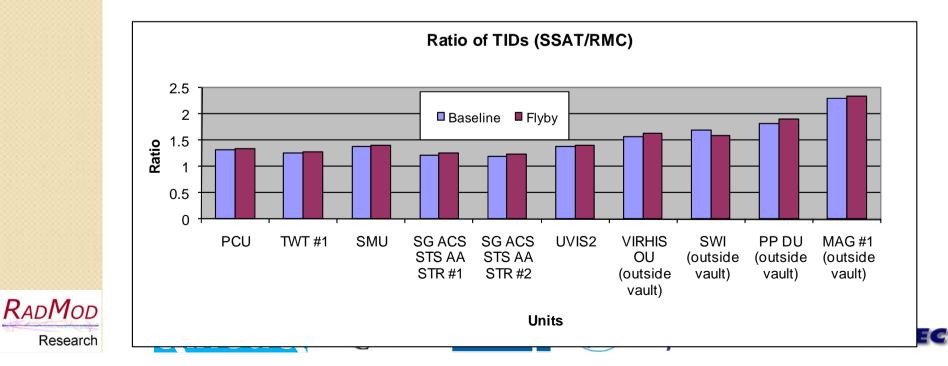




Comparison with SSAT Results

SSAT analysis were carried out for the same locations with the same geometry model. The calculated TID in krad(Si)

		Baseline	Flybys
Platform	PCU	90.69	112.83
	TWT #1	92.27	113.53
	SMU	90.83	113.03
	Star Tracker 1	111.52	138.06
	Star Tracker 2	111.2	138.17
Payload	UVIS2	89.53	111.64
	VIRHIS OU	197.39	237.55
	SWI	225.53	248.87
	PP DU	82.52	103.52
	MAG #1	163.01	204.89



Future Developments

- A new team has been assembled in response to new ESA ITT (CIRSOS): New Integrated Modelling Environment (IME)
 - collaborative and iterative modelling approach, operational s/w for CVMs
- Geometry manager
 - Model configuration tool: GDML modular schema
 - New visualisation tool
- Application tools
 - Internal Charging (IC) analysis
 - Based on the ELSHIELD work
 - Build-in libraries of materials and components at risk
 - SSAT
 - 2-stage analysis approach
 - FMC and RMC in both stage
 - General parametric analysis
 - Template based solution
- Simulation manager
 - Utility to use commercial cloud computing facilities, e.g. EC2
- Lots of enhancement to post-processing and many more...





Backup slides





Interactive Python console/ Python

editor

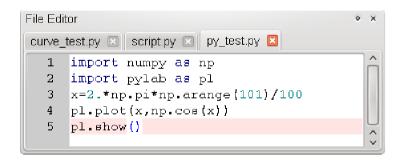
Post Processina o x Results Viewer Python console Plot Commands Us <> Running... E, A 00:26:03 📕 Variables ->>> import pylab as pl >>> x=2.*np.pi*np.arange(101)/100 >>> pl.plot(x,np.cos(x))[<matplotlib.lines.Line2D object at 0x278e050>] >>> pl.show() import numpy as np import pylab as pl x=2.*np.pi*np.arange(101)/100 pl.plot(x,np.cos(x)) pl.show() [<matplotlib.lines.Line2D object at 0x256fa50>] >>> >>> >>> >>> >>> >>> >>> >>> >>> import numpy as no >>> import pylab as pl >>> x=2.*np.pi*np.arange(101)/100 >>> pl.plot(x,np.cos(x)) [<matplotlib.lines.Line2D object at 0x278eed0>] >>> pl.show()

Python script editor

- Syntax recognition based on QScintilla
- menu for run, save, edit, add commands

Interactive Python console

- Using the Spyder python library
- Run of user script from popup menu of the PYTHON editor
- Visualisation of python command send by the GUI
- Pre imported modules : numpy, pylab,
 DataManager, function loading



Reading/Viewing/Access of simulation results

1.0

Post Processing		♦ X
Results Viewer Pytho	n console 📗 Plot Commands 📗 User Functions 🗌	
Name	i type	value
GRAS results CSV files		
- sim1.csv - dose1	GRAS sim results DOSE	8.8981
L. DOSE 5	PECTRUM Histogram 1D	
general	COMMON	

Reading/Viewing of simulation results

- PYTHON interface to Spenvis CSV C++ code to read CSV files
- Direct View of GRAS CSV file in Editor
- Results viewer widget as a file structure viewer

Access of data in user script

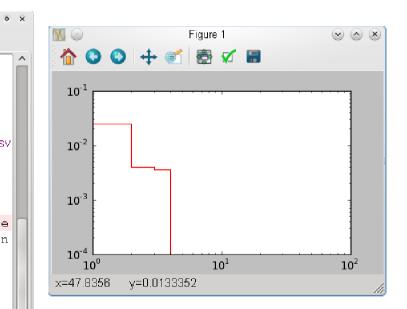
- Access of DataManager in user script
- Python Cmds to acess Histograms directly added in the script from a popup menu of the results viewer

File Editor curve_test.py 🎆 script.py 🐺 data.py 🗵 sim1.csv(read only) 🎆 2 theDataManager=AnalysisDataManager() #Load GRAS csv file sim1.csv 3 theDataManager.load gras csv file(115,data name='sim1.csv') 4 theHisto=theDataManager.list_of_gras_csv files['sim1.csv']['histo1[5 6 xlow vec=theHisto['ower'] xup vec=theHisto['upper'] 7 8 v vec=theHisto['value'] verror vec=theHisto['error'] 9

Plotting of simulations results

Post Processi	ng	ô x
Results Viewe	er Pyth	on console Plot Commands User Functions
Plot	Figure	Add Plot Cmds To Script
📄 Sa	ve Figure	Filename
Figur	e Size	
dX	8,00 🗘	dY 9,00 ♦ cm ~
Divid	e Figu	ire in sub-plots
🔽 ac	File Edit	or 🄹
X-axis	curve_	test.py 🐲 script.py 🀲 plotdata.py 🔼 sim1.csv(read only) 🐲
	21	****
Curv	22	pl.subplot(1,1,1)
Select	23	#Add curves
	24	
Color	25	pl.subplot(1,1,1)
Select	26	theHisto=theDataManager.list_of_gras_csv_files['sim1.csv
	27 28	xlow_vec=theHisto['lower'] xup_vec=theHisto['upper']
Axis	20 29	y_vec=theHisto['value']
Select	30	yerror_vec=theHisto['error']
🖓 Lo	31	<pre>xv=np.array ([xlow_vec, xup_vec]).transpose().reshape</pre>
	32	<pre>yv=np.array ([y_vec, y_vec]).transpose().reshape((len</pre>
🗸 Lo	33	pl.plot(xv,yv+1.e-250,'red')
	34	#Axes Subplots
L	35	****
	36	<pre>axes=pl.subplot(1,1,1)</pre>
	37	axes.set_xscale("log")
	38	axes.set_yscale("log")
	39	<pre>ymax=axes.get_ylim()[1]</pre>
	40	ymin=ymax/1.0000e+03
	41	<pre>axes.set_ylim((ymin,ymax)) """""""""""""""""""""""""""""""""""</pre>
	42 43	#Show plot in interactive window #################
	4-3 - 4.4	

- Direct plotting of single ID histogram
- Multi curve plottings using a plotting widget
- Interactive adding of plotting cmds from plotting widget to user script



Response function analysis

Project Viewer	ō	>
Name	type].
÷- test	project	
+ Geometry	geo	
+ Environment	env	
- Simulations		
+ sim1	simulation	
- response sim	simulation group	
	GRASdoubles response function file	
+ response sim Ekn1.5843e-01 2.5119e-01MeV	simulation	
+ response_sim_Ekn1.0000e-01_1.5849e-01MeV	simulation	
+ response_sim_Ekn3.9811e-01_6.3096e-01MeV	simulation	L
response sim Ekn1.0000e-02 1.58496-02MeV	simulation	

•<u>Signals vs primary spectrum</u>

- Convolution of response functions with user selected spectra
- Use of SciPy library
- ASCII table of scalar signals in function of user selected spectra

Response functions vs primary energy

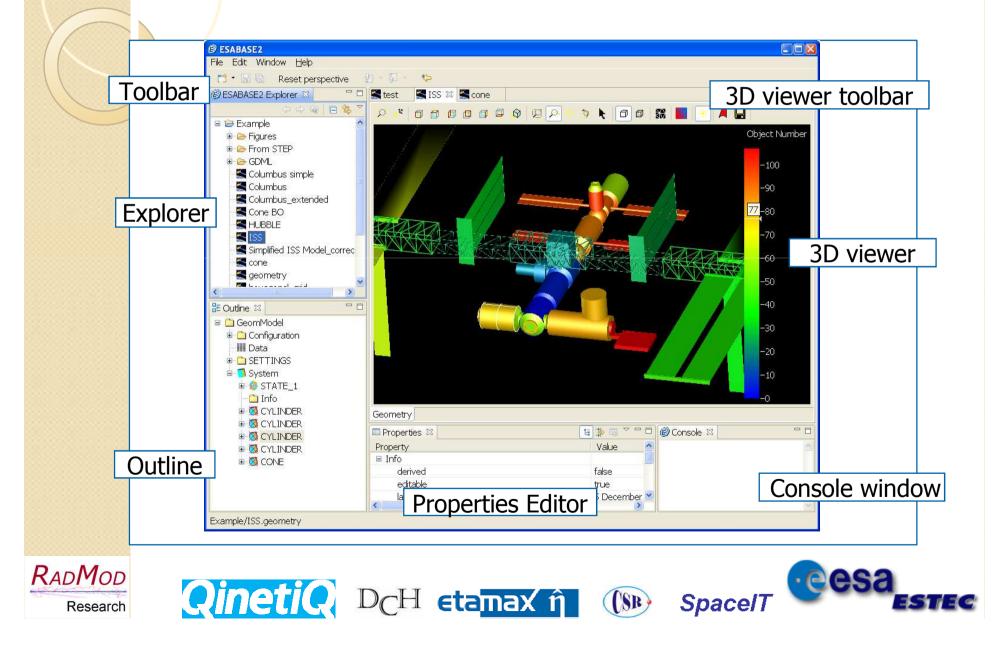
- Multiple simulations vs primary energy
- Log or linear energy bins
- General concept of simulation group for all type of parametrized simulations
- End of simulations: production of an ASCII table containg all scalars (TID,NIEL,...) in function of primary energy bins

File Edit	tor • x				
3 resp	response_sim_stat_double_for_selected_spectra.txt(read only) 🗵 < >				
1	#Convolution of response function response_sim_stat_dc				
2	#line 1: spectrum newEle2.dat				
3	#line 2: spectrum newEle1.dat				
4	#column 1: module dose Dose_value				
5	#column 2: module dose Dose_error				
6	2.5689e+13 3.1352e+10	ł			
7	2.5689e+13 3.1352e+10				
	×	<u>_</u>			

Definition/loading of user functions for radiation effect analysis

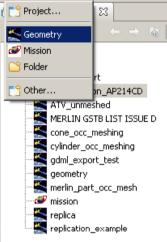
Post Processing		♦ X		
Results Viewer Python cons	ole Plot Commands User I	Functions	 Definition of user functions from the GL 	Jl or in
Name	expression	:	PYTHON script	
- from expression - f1 - f2 - f3 - f4 - parameterized - from table	x+2.*x*× × np.sin(x) f3+cos(x)		 User functions stored in the database Access of user functions in PYTHON so 	ripts
🔜 💿	RestSIM.py	0 0		
			File Editor	♦ X
function category create function	name f5 DisplacementDamage >	expression = exp(-x/3.789)	<pre>script py(read only) Curve_test.py 4 5 curve=GetDDamageCurve ("JPL_NRL_NASA_Si_proton") 6 x=curve ["ekin_vec"] 7 y=curve ["NIEL_vec"] 8 y1=curve ["function"] (x) 9 10 f1=GetCurveFunction ("f1") 11 f2=GetCurveFunction ("f2") 12 13 14 15 16 17</pre>	

ESABASE2 GUI Overview

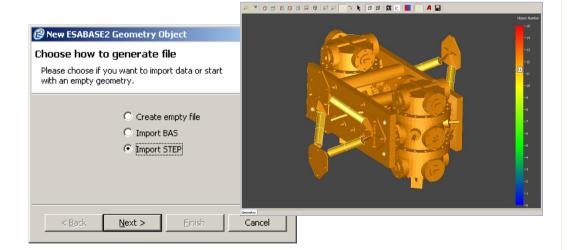


STEP AP203/214 Import

- Import of any STEP AP203/214 files, also of high complexity.
- Tested with a large number of • files originating from various CAD tools
- Accessible via the respective open file menu or toolbar entry.
- FSABASE2 📑 🗝 🔚 🕼 Reset perspective File Edit Window Help 😚 Project... New Project... X Onen File. Keometr 🍝 Geometry Close Ctrl+W 🥔 Mission Close All Ctrl+Shift+W Folder 🥩 Mission Save Cher... Folder Save As.. Save All Other... Convert Line Delimiters To ATV unmeshed Switch Workspace cone occ meshina 🌄 cylinder_occ_meshing 1 ATV@Station AP214CD.geometry [Example] 2 MERLIN GSTB LIST ISSUE D.geometry [...] 🌉 qdml export test 3 gdml_export_test.geometry [Example] aeometrv 4 geometry.geometry [Example] 💒 merlin_part_occ_mesh Exit 🛷 mission 🌉 replica



- Opens a file selection dialogue
- Can take a while, depending on the complexity of the geometry...

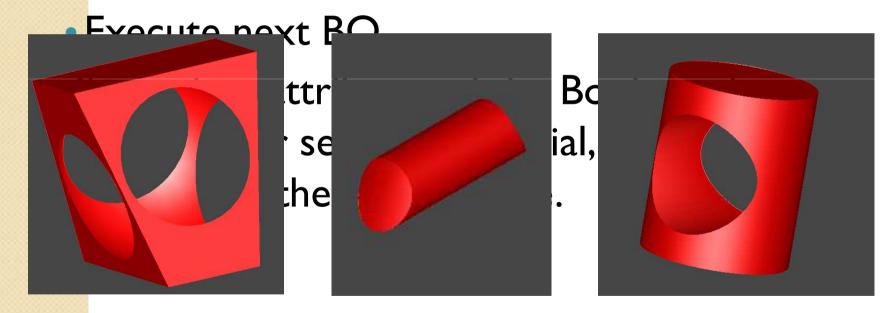


Boolean Operations (I)

- Three types of Boolean Operations (BO) implemented:
- intersection union subtraction After the operation is performed, the child shape is stored under the BOP-nod It can be edited via the shape Any changes made are applie Boolean shape. If a BO is removed from the BOP node, the operation is u

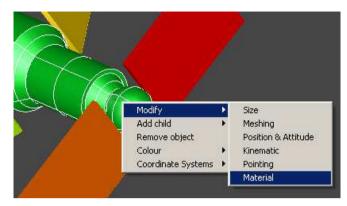
Boolean Operations (2)

- Series of Boolean Operations can be performed following the defined workflow:
 - Add child to Boolean shape parent



Material Editor

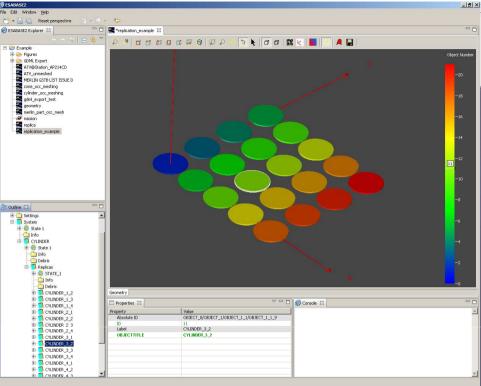
- Objective: Assignment of materials to geometrical objects.
- Underlying comprehensive material database.
- **P**art of the shape wizard.
- Accessible via
 - the Outline context menu,
 - the 3D view context menu ('Modify --> Material').
- Allows
 - editing of material properties,
 - creation of user defined materials.
- Materials can be assigned only, if they are copied from the material database to the geometry.
- The elements a material is consisting of are automatically copied to the geometry, when the material is copied to the geometry.



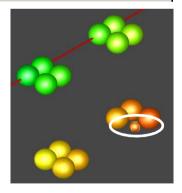
🝘 Shape Wizard				×
Material				P
				9
Size Meshing Position & a	Attitude Kinemal	tic Pointing Mate	rial	
Assign materials				2
Whole Shape	LUMINUM		Propagate	
	Surface	Material		
-	347460			
C Replace	~	with	Propagate	
-		. ,		
Material overview				
		(
ALUMINUM	Atm/Ion		GDML Used 🔺	
GOLD				
GaAs				
	Edit mate	rials		
ļ				
		< Back Next	> <u>E</u> inish	Cancel

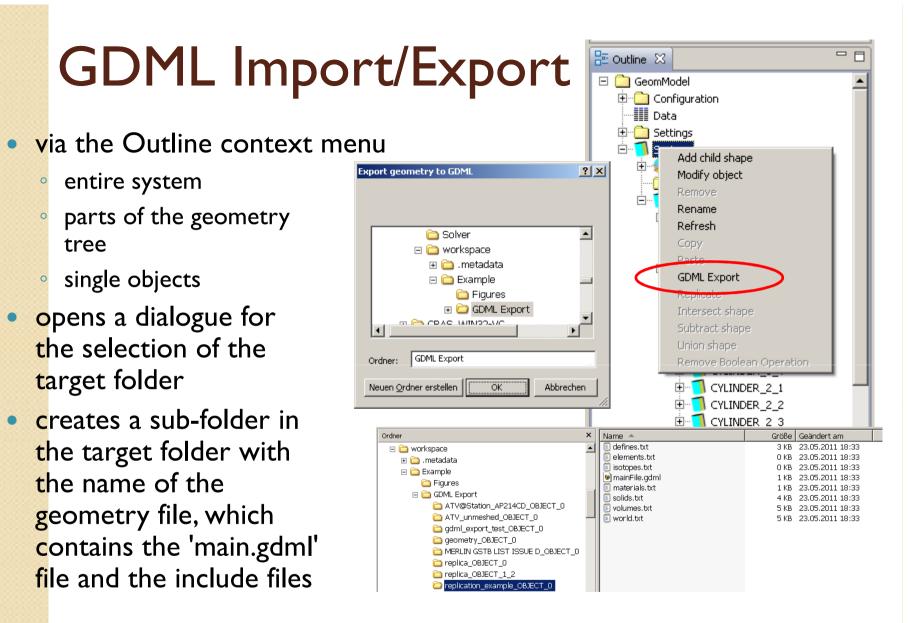
Replicated Structure

- Objective: enable the replication of geometrical objects in x- and y-direction with user defined different off-sets.
 - Example:
 - replicated cylinders
 - 5 in x-direction
 - 4 in y-direction
 - different off-sets in x- and y-direction
 - index of the replica is given in its name
 - here: cylinder_3_2 :
 - 3rd in x-direction
 - 2nd in y-direction
 - Replication of the parent object of a replicated structure replicates the entire structure.
 - The properties each object of the replicated structure can be edited individually.



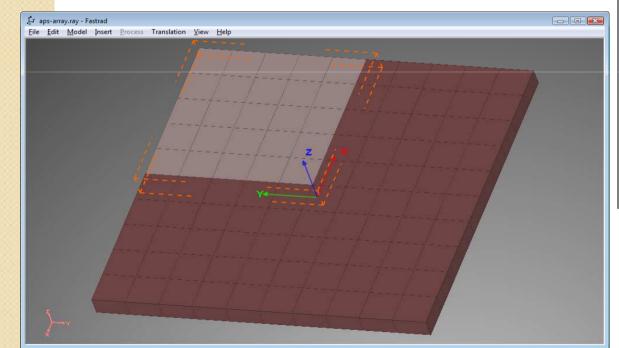


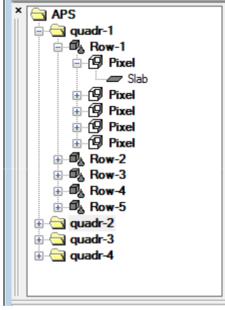




JUICE Geometry: Pixellated APS Model - FASTRAD

- Repeated APS pixels can be modelled using the combined functions of copy/paste and transformation
- But volume names have to be changed manually
 - This is required if each pixel needs to be identified uniquely
 - Impractical for large arrays such as the APS (1024x1024)





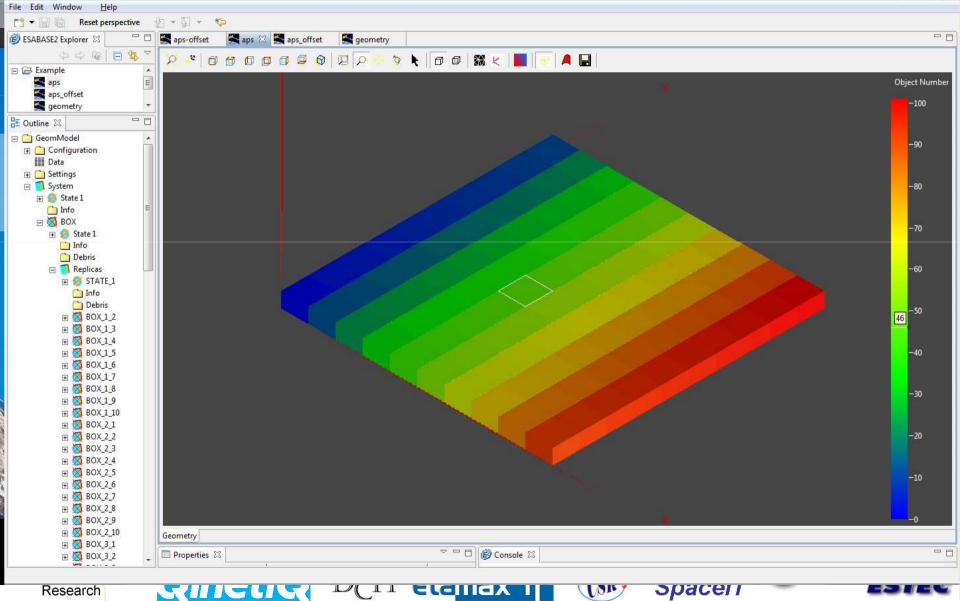








JUICE Geometry: Pixellated APS Model – Open Frontier E ESABASE2



Research

- @ X