

MCICT - MONTE-CARLO INTERNAL CHARGING TOOL

Fan Lei⁽¹⁾, David Rodgers⁽²⁾, Pete Truscott⁽³⁾

⁽¹⁾RadMod Research, Camberley, Surrey, UK, Email: flei@radmod.co.uk

⁽²⁾ESTEC/ESA, Noordwijk, The Netherlands, Email: david.rodgers@esa.int

⁽³⁾Kallisto Consultancy, Farnborough, Hampshire, UK, Email: prtruscott@kallistoconsultancy.com

ABSTRACT

Monte-Carlo Internal Charging Tool, MCICT, is a new 1D internal charging analysis code developed within the ESA JUICE Charging Analysis Tool project. What separates it from other similar tools is the detailed treatment of charged particle transport based on the Geant4 toolkit. This allows it to be applied to all charged particle radiation environments, and with multilayered geometry of any materials. In this paper we report on its main features, and the comparison studies with other codes.

1. INTRODUCTION

The extreme internal charging environment faced by the European Space Agency's Jupiter Icy Moons Explorer (JUICE) [1] is posing a number of new challenges to internal charging analysis. Robust, sophisticated 1D analysis tools are required, and there are also the emerging requirements for 3D-based analysis. In response to these new requirements, for the JUICE development in particular, recently under ESA funding a new 3-D Monte-Carlo internal charging code [2, 3] has been developed. However, from the engineering point of view, the 3-D code is too slow for the repeated simulations required in the planning phases and initial assessments of missions and it is here that 1-D codes are most useful. Also there is the need for rapid and accurate 1D code in the derivation of dielectric material parameters from experimental data. For these reasons, ESA has updated the DICTAT code with improved algorithms and with extended applicable energy range [4]. However, even the new DICTAT tool can only treat geometries such as a single shielding plate and a single dielectric plate, whereas for JUICE charging analysis there are requirements for multi-layered shielding and dielectrics configurations. Hence there is a the need for the development of a new 1D Monte Carlo based internal charging analysis tool, within the JUICE Charging Analysis Tool - JCAT project [5].

2. MCICT DESIGN AND IMPLEMENTATION

The internal charging analysis can normally be carried out in two parts: first the calculation of charge and dose deposition in the dielectric and then the calculation of the electric field, i.e. solving the circuitry equations. For example, in the DICTAT and NUMIT [6] codes, the charge and dose depositions due to the incident

radiation are calculated using empirical analytical formulas, and the electric fields are calculated using solvers based on Ohm's law.

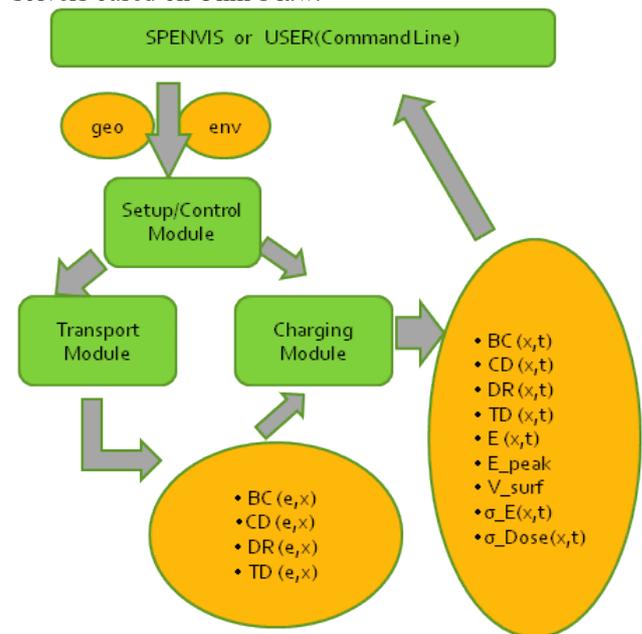


Figure 1: An overview of the MCICT software design.

The design and implementation of MCICT also follows the same principle. In Fig.1 is the top-level design of this tool. It consists of three modules: setup/control, Monte-Carlo particle transport and an electric solver. The setup/control module constructs the geometry setup for both the transport and charging modules. The particle transport module provides the electric solver the required dose and charge distributions in the geometry. The design also shows when the code is operating within the SPENVIS [7] system that provides the GUI for a user to define the geometry, irradiation conditions and the type of analysis.

2.1 The transport Module

While DICTAT uses empirical algorithms in calculating the charge and dose depositions, a 1D tool based on detailed Monte-Carlo simulation of the particle transport in the shields and dielectric can meet all the new requirements in the analysis of sophisticated layered structures. The Monte-Carlo code for radiation dose analysis, GRAS [8] which has been widely used in

radiation effect analysis in many ESA projects has proved to be a very fast and easy to use tool. An updated version of GRAS is used in MCICT to perform rapid simulation of particle transport in 1D geometry setups and generate the dose and charge deposition distributions required for the charging analysis stage.

For time-variable environments, a built-in response function approach is implemented. MCICT will first generate a response function of the specified geometry to the given environment, by performing a series of GRAS simulations as a function of particle energy. With the response function, the time dependent charge and dose distributions can be calculated by convolution with the given incident charged particle specifications.

2.2 The Charging Module - Circuit Solver:

For a simple 1D case it is possible to calculate the electric field iteratively following Ohm's law. For example, in DICTAT, the electric field is calculated as for a simple parallel capacitor:

$$E(t) = J/\sigma (1 - \exp(-t/\tau))$$

where $\tau = RC = \sum \epsilon_i / \sigma_i$, assuming the dielectric is divided into multiple layers.

A PYTHON version of the DICTAT field solver has been implemented in MCICT.

2.3 Summary of the Main Features

- 1) Geometry modelling:
 - Arbitrary layers of shields and dielectrics and grounding conductors.
 - Slab or cylindrical shape
- 2) Environment specification:
 - All types of charged particle or photons treated, e.g.: e-, p and heavy ions, and X-rays
 - Unidirectional or isotropic angular distribution
 - A selection of 9 options for temporal and spectral distributions, from constant monochromatic energy, to both time-dependent spectrum shape and intensity.
- 3) Analysis/Outputs:

The GRAS transport module generates the charge and dose distributions due to the incident radiation in the geometry set up in mesh format. The field-solver outputs the calculated dose rates, the charging current, the electric field strength and the dielectric conductivities, all in histograms defined by the sub-layers of the geometry set up and as a function of the observation times. If running in

batch mode, plots of these histograms are also produced in png format.

2.4 Example of the Input File

The execution of MCICT is controlled by an ASCII input file. An example input file is shown in Fig.2 of this section and it consists of 5 blocks, each marked with a unique head line such as “__Env__”. Lines which start with the symbol “#” are treated as containing only comment and are ignored by the file parser.

The Env block:

The first line entries are:

particle, Energy_unit, Flux_units, angle_type

- particle: options are e-, proton, or any valid Geant4 particles
- Energy_unit: energy units of the incident particle, options are "MeV", "keV", or all valid Geant4 energy units
- Flux_units: integral flux of the incident radiation. It can be "particle/cm2/s" only for now.
- angle_type: directional distribution of the incident particles. Options are "normal" or "iso"

The remaining lines in this block define the distributions of the incident particles. There are 9 options, but only one can be used at a time.

Option 1: Constant Mono-energetic source
Mono, Energy, flux

Option 2: Constant power-law spectrum
Pow, alpha, Emin, Emax, flux

Option 3, Constant Exponential spectrum
Exp, E0, Emin, Emax, flux

Option 4, Constant Arbitrary (point-wise) spectrum
Arb, E1, ..., En, F1, ..., Fn, flux

Option 5: Mono-energetic, time-variable intensity
Mono, Energy
t1, f1
..
tn, fn

Option 6: Power-law, time-variable intensity
Pow, alpha, Emin, Emax
t1, f1

..
tn, fn

Option 7, Exponential, time-variable intensity

Exp, E0, Emin, Emax

t1, f1

..
tn, fn

Option 8, Arbitrary, time-variable intensity

Arb, E1, .., En, F1, .., Fn

t1, f1

..
tn, fn

Option 9, Arbitrary, time variable spectrum shape and intensity

t1,f1, E11, .., E1m, F11, .., F1m

t2,f2, E21, .., E2m, F21, .., F2m

..
tn,fn, En1, .., Enm, F21, .., Fnm

All times (t_i) should be given in seconds, flux (f_i or *flux*) is the integral omni-directional flux between *Emin* and *Emax*, or *E1* and *En*.

The Geo Block:

The first line is to specify the geometry shape: slab/cylinder, e.g.

slab

The following lines define the layer composition. One line per layer:

layer_name, thickness, material_name, temp, ground, n-sub

- *layer_name*: name of the layer
- *thickness*: layer thickness in mm
- *material_name*: the material of the layer. It has to be one of the Geant4 predefined materials, e.g. *G4_Al*, or one of the materials defined in the *__Mat__* block (below).
- *temp*: the temperature of the layer in K
- *ground*: the grounding condition of the layer. Options are "G" for ground, "F" for floating conductor and "D" for dielectric layer.
- *n-sub*: the number of sub-layers to be divided into.

The Mat Block:

This block defines new user materials, e.g.

MyWater H2-O 1.0

- first entry is the name of the new material, i.e. *MyWater*
- second entry is its chemical composition, either given in molecule composition, i.e. *H2-O* for water,

or in fraction-of-mass by elements, i.e. *H0.111-O0.889* for water

- third entry is the density of the material in g/cc

The Die Block:

This block defines the dielectric parameters of the dielectrics used:

name, permittivity, dark_conductivity, kp, delta, ea

The definition and units for the dielectric parameters are the same as that of DICTAT. For example:

G4_MYLAR, 4.70, 1.48e-18, 9.84e-14, 1.77, 0.42

G4_TEFロン, 2.31, 9.31e-16, 4.0e-15, 0.61, 0.81

The Obs Block:

The first line is to specify the times the electric field in the dielectric layer(s) should be measured/calculated. The times can be given at fixed intervals or in a table of given times. All time are specified in seconds

step/table, times ..

e.g.

step, 1800, 144000, 1800

This specifies the observation time is from 1800 seconds and ends at 144000 seconds, in steps of 1800 seconds.

The next line is to allow interactive plotting or not

batch, true/false

When *true* the execution is in batch mode and all plots are saved automatically in .png files. When *false*, interactive plotting is activated and the user will be able to manipulate the plotting process.

The last line of the input file is to specify the number of incident particles for the GRAS simulation, e.g.

events, 100000, 10000, false

The second entry specifies the number of events to be executed for each GRAS run. The third entry is the event interval for which GRAS result should be outputted. The last entry can be false or true. When false, the GRAS simulation will not be executed and MCICT will use existing GRAS results for the field calculation. This allows user for example to change the dielectric parameters and recalculate the field without the need to run GRAS simulations.

```

__Env__
# 9 options are available
e-, MeV, /cm2/s, iso
Pow, -0.5, 0.5, 5.
o., 1e6
100., 1e7
1800., 1e5

__Geo__
# slab/cylinder
cylinder
# the first layer is at the outer most!
# layer_name, thickness, material, temp, D/G/F, n-sub
# (thickness in mm, temp in K, D: dielectric; G: ground; F: floating
# metal)
L1, 0.1, G4_Al, 110, G, 1
L2, 0.2, G4_MYLAR, 110, D, 20
L3, 0.1, G4_Cu, 110, F, 1
L4, 0.2, G4_TEFLON, 110, D, 20
L5, 0.1, MyAl, 110, G, 1

__Mat__
# MULASSIS format
MyAlAl 2.0

__Die__
# name, permittivity, dark_conductivity, kp, delta, ea
#(see README for definition and units)
G4_MYLAR, 2.31, 9.31e-16, 4.0e-15, 0.61, 0.81
G4_TEFLON, 2.31, 9.31e-16, 4.0e-15, 0.61, 0.81

__Obs__
#step/table
# step, start time, end time, nr of steps (time in seconds)
step, 60, 3600, 60
# plotting (true) or not (false)
batch, false
#
# events, output_mod, number of events, true/false - run GRAS?
events, 100000, 100000, true

```

Figure 2: Example MCICT input file for power-law spectrum electrons incident upon a cylindrical structure of aluminium, Mylar, copper and Teflon.

2.5 Examples of the Outputs:

The execution of MCICT with an input file named xxx.inp will result in the generation of a large number of output files in the current directory. Among them the main results are stored in the following ones:

- xxx-efield.csv: the calculated electric field distribution, at specified observations times

- xxx-doserate.csv: the calculated dose rate, as a function of observation time
- xxx-chargerate.csv: the charging current as a function of observation time
- xxx-current.csv: the follow current as function of observation time

These files are saved in the SPENVIS CSV format, as MCICT has been successfully integrated into the SPENVIS systems. This also makes it easier for them to be imported and re-processed with another analysis tool.

MCICT will also plot the results either in batch or interactive mode. Fig.3 and 4 are examples of these plots.

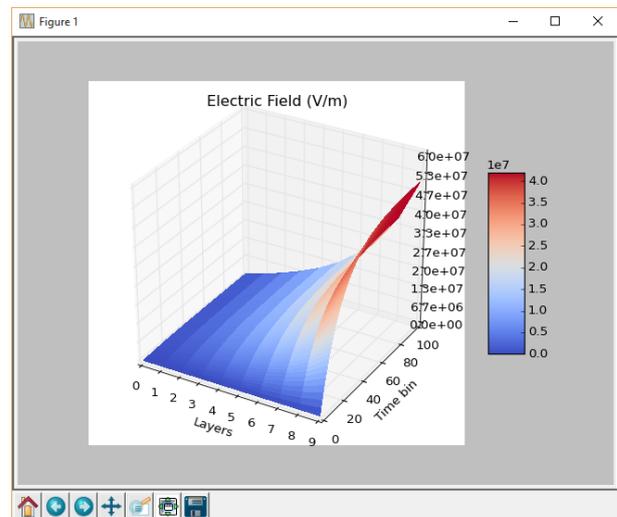


Figure 3: An example of the MCICT calculated electric field strength in each sub-layer of the dielectric(s), as a function of observation time.

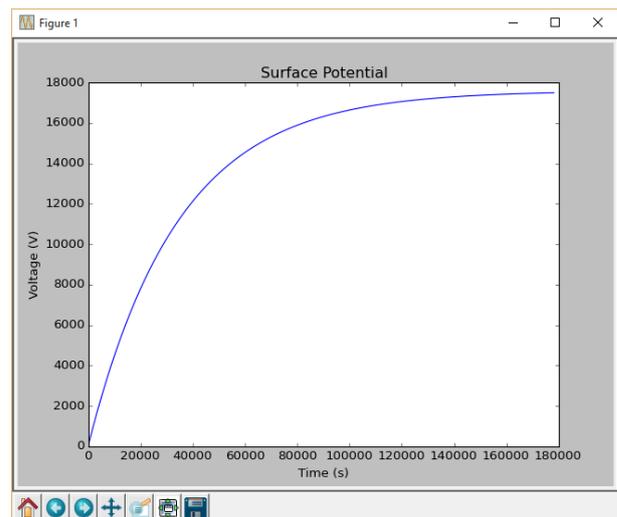


Figure 4: An example of the MCICT calculated top surface potential of a single dielectric layer as a function of observation time, assuming the layer is grounded at the back side.

3. COMPARISONS WITH OTHER CODES

Comparison studies of MCICT with two similar codes, DICTAT and NUMIT, have been carried out as part of its development. In this section we show the results for a simple sample of 1 PMMA layer irradiated with 500 keV electrons. The details of the set up are:

- 1 mm PMMA, divided into 10 sub layers, grounded at the back side only
- Conditions and material parameters are outlined in Tab. 1
- Assuming the power index for radiation induced conductivity $\Delta=1.0$

Table 1: The main parameters used for the comparison studies. This test case is based on the sample problem in [9]

INPUTS TO NUMIT FOR THE SAMPLE PROBLEM			
J_{inc}	electron beam current	-1.0×10^{-7}	A/m ²
T_0	electron beam energy	0.5	MeV
d	dielectric thickness	0.001	m
Z_{eff}	effective atomic number of the dielectric	6	
A_{eff}	effective atomic mass of the dielectric	12	
ϵ	dielectric permittivity at 300°K	3.28×10^{-11}	F/m ($\kappa = 3.71$)
ρ	dielectric's density	1.19	g/cm ³
g_{dark}	dark conductivity	1.0×10^{-16}	ohm ⁻¹ m ⁻¹
g_{rad}	radiation induced conductivity	2.76×10^{-16}	sec ohm ⁻¹ m ⁻¹ rad ⁻¹

To setup the simulation in NUMIT 2, it is necessary to inset a 1m air gap between the front electrode and the PMMA layer to make it back side grounded only. The PMMA is sliced into 10 sub layers, and the derived parameters per sub layer are the average at front and back boundaries of the layer. For the DICTAT dose-rate and beam current, they are calculated analytically using the formulas given in its manual.

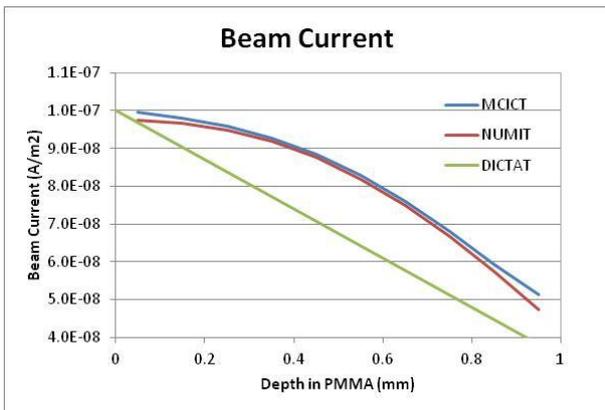


Figure 5: The calculated beam current as a function of the depth in the PMMA layer. There is a good agreement in general, between MCICT and NUMIT in particular.

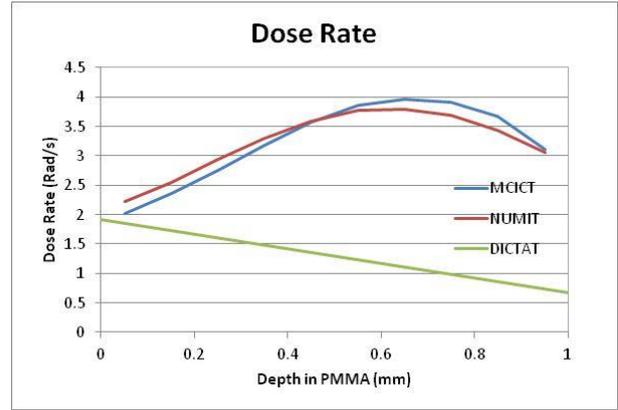


Figure 6: The calculated dose rate as a function of depth in the PMMA layer. Again there is a good agreement between MCICT and NUMIT, but some discrepancy with the simple algorithm used in DICTAT.

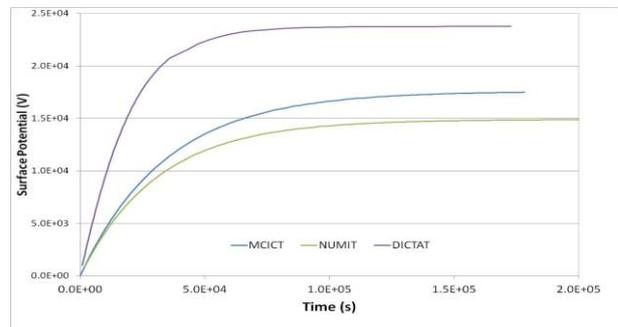


Figure 7: The calculated top surface potential as a function of the exposure time. There is a good agreement in general.

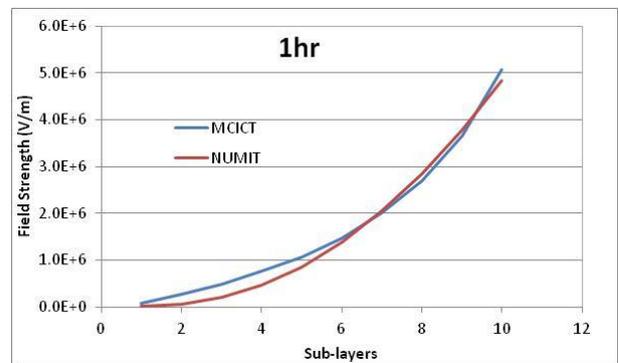


Figure 8: The calculated electric field strength at the 1 hr exposure time in each of the PMMA sub layers.

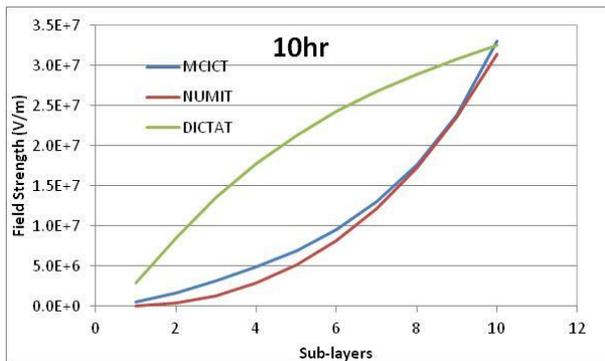


Figure 9: The calculated electric field strength at the 10 hrs exposure time in each of the PMMA sub layers. There is a very good agreement between the MCICT and MUNIT values. However the DICTAT results are different for the middle sub layers.

4. CONCLUSIONS AND DISCUSSIONS

A 1D internal charging analysis tool, MCICT, based on detailed MC simulation of charged particle transport has been developed. It is capable of treating all types of charged particle incident onto a multilayered slab or cylindrical geometry of different materials, including dielectrics, shielding and grounding metals. The code has been successfully verified with other established codes. The performance of the codes is acceptable for most engineering applications, with simulation time typically in seconds on standard desktop computers. MCICT is available as standalone applications for both Windows and Linux operating system. It has also available on the internet as it has been integrated in the SPENVIS system.

Further verification and validation of the tool will be carried out, for example comparison with the 3D internal charging tool SPIS-IC/CIRSOS. Future developments of MCICT include the use of a finite element/volume based electric field solver is planned. Also it is possible turn MCICT into a full 3D tool. In fact the GRAS based charge particle transport module is already a full 3D code. What's needed is a 3D field solver, and we are evaluating two open-source libraries, FIPY [10] and MEEP [11], both look to be very suitable candidates.

5. ACKNOWLEDGEMENTS

The JUICE Charging Analysis Tools Project is sponsored by the European Space Agency under the Technology Research Programme, contract number 4000109999/13/NL/AK.

6. REFERENCES

1. "JUICE Exploring the emergence of habitable worlds around gas giants," ESA Assessment Study Report ("Yellow Book"), ESA/SRE(2011)18, December 2011.
2. P. Sarrailh et. al.; "Development and Validation of a 3D Time Dependent Model used for the Simulation of Internal Charging at Spacecraft Level", Proceeding of the 14th Spacecraft Charging Technology Conference, April. 2016.
3. SPIS-IC/CIRSOS: <http://dev.spis.org/projects/spine/home/spis>
<http://spitfire.estec.esa.int/trac/CIRSOS/wiki/SUM/Introduction>
4. David J. Rodgers, Fabrice Cipriani, John Sorensen and Christian Erd, "DICTAT Internal charging Code for the Jovian Environment", Proceeding of the 12th Spacecraft Charging Technology Conference, Sep. 2012.
5. Pete Truscott et al, "JUICE Charging Analysis Tools Project: Contract Final Report," Kallisto Consultancy Ltd, KALLISTO/CR/15008, 2016.
6. NUMIT 2.0: <http://www.openchannelfoundation.org/projects/NUMIT2.0/>
7. SPENVIS – <https://www.spennis.oma.be/>
8. G. Santin, V. Ivanchenko, H. Evans, P. Nieminen, E. Daly, "GRAS: A general-purpose 3-D modular simulation tool for space environment effects analysis", IEEE Trans. Nucl. Sci. 52, Issue 6, 2005, pp 2294 – 2299. <http://space-env.esa.int/index.php/geant4-radiation-analysis-for-space.html>
9. Insoo Jun, Henry B. Garrett, Wousik Kim, and Joseph I. Minow, "Review of an Internal Charging Code, NUMIT", IEEE TRANSACTIONS ON PLASMA SCIENCE, VOL. 36, NO. 5, OCTOBER 2008, 2467.
10. J. E. Guyer, D. Wheeler & J. A. Warren, "FiPy: Partial Differential Equations with Python," Computing in Science & Engineering 11(3) pp. 6–15 (2009), doi:10.1109/MCSE.2009.52, <http://www.ctcms.nist.gov/fipy>
11. Ardavan F. Oskooi, David Roundy, Mihai Ibanescu, Peter Bermel, J. D. Joannopoulos, and Steven G. Johnson, "MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method", Computer Physics Communications 181, 687–702 (2010). <http://ab-initio.mit.edu/wiki/index.php/Meep>