



Enhanced Orbit Propagator

ESA Contract No. RFP/D/IPL-PTE/GLC/al/557.2014

NEOPROP2 Software User Manual



Document Number:	ASTOS-NEOPROP2-SUM-001		Date:
Issue:	1.2		2015-11-23

	Name/Function	Organization	Signature	
Prepared by:	V. Zuccarelli	Astos Solutions		2015-11-23
Checked by:				
Project Management:	V. Zuccarelli	Astos Solutions		2015-11-23

Document Change Record

Issue	Date	Affected Chapter/Section/Page	Reason for Change Brief Description of Change
1.0	2015-05-22	All	First issue
1.1	2015-09-11	9.2, Appendix B	Fixed few typos, extended section 9.2 with Linux dependencies and added Appendix B
1.2	2015-11-23	6.3.2, Added chapters 11.2.1.3, 11.2.2.3, 11.2.3.3, 11.2.3.4	Added Step-by-Step procedures and Scenario Import section

Table of Contents

1	Introduction.....	6
2	Applicable and Reference documents	7
2.1	Applicable Documents	7
2.2	Reference Documents	7
3	Terms, definitions and abbreviated terms	9
3.1	Acronyms	9
4	Conventions	10
5	Purpose of the Software	11
6	External view of the software	12
6.1	Executable Files.....	12
6.2	Libraries	12
6.3	Input/Output Files.....	17
7	Operations environment	28
7.1	General	28
7.2	Hardware configuration.....	28
7.3	Software configuration	28
7.4	Operational constraints	28
8	Operations basics.....	29
9	Operations manual	30
9.1	General	30
9.2	Set-up and initialization.....	30
9.3	Getting started	30
9.4	Normal operations.....	31
9.5	Normal termination.....	32
9.6	Batch File	32
9.7	Error conditions.....	33
9.8	Recover runs.....	33
10	Reference manual	34
10.1	Introduction	34
10.2	Help method.....	34

10.3	Screen definitions and operations	34
10.4	Commands and operations.....	34
10.5	Error messages.....	34
11	Tutorial	35
11.1	Introduction	35
11.2	Using the software on a typical task	35
	Appendix A – Reference Frames.....	57
	Appendix B – List of implemented algorithms.....	58

Table of Figures

Figure 1 Orbit determination algorithm diagram	37
Figure 2 Propagation XML Node.....	41
Figure 3 Final Epoch (Julian Date) XML Node	43
Figure 4 Third body trajectory flag node	44
Figure 5 Asteroid XML Node.....	44
Figure 6 PPN Relativistic Effects Model Node	47
Figure 7 Virtual Asteroids XML Node.....	48
Figure 8 Number of threads XML Node	48
Figure 9 GUI's screenshot and structure	51
Figure 10 Screenshot of the GUI dialog to create a new scenario	53

1 Introduction

This document is the Software User Manual of NEOPROP2. It describes the different packages of the software, how they can be obtained and installed, and it summaries the possible settings and output.

It has been prepared for the Project *Enhanced Orbit Propagator* performed by Astos Solutions under contract of the European Space Agency.

.

2 Applicable and Reference documents

2.1 Applicable Documents

- [AD1] TEC-ECM-SOW/2010.0915, SoW for "Orbit Propagation Algorithms for the Space Situation Awareness"
- [AD2] TEC-ECM-SOW/2014.0305, SoW for "Enhanced Orbit Propagator"
- [AD3] ASTOS-NEOPROP-TN-001 – Orbit propagation algorithm for the Space Situational Awareness - Review and Assessment performed
- [AD4] ASTOS-NEOPROP-TN-003 – Orbit propagation algorithm for the Space Situational Awareness - New approaches for the computation of orbit propagation
- [AD5] ASTOS-NEOPROP-TN-004 – Orbit propagation algorithm for the Space Situational Awareness - Trade-off Study
- [AD6] ASTOS-NEOPROP2-TN-001 – Review of NEOPROP Limitations and Identification of new Algorithms
- [AD7] ASTOS-NEOPROP2-SRD-001 – Software Requirements Document
- [AD8] ASTOS-NEOPROP2-SDD-001 – Software Design Document
- [AD9] ASTOS-NEOPROP2-ICD-001 – Interface Control Document
- [AD10] ASTOS-NEOPROP2-SVVP-001 – Software Verification and Validation Plan
- [AD11] ASTOS-NEOPROP2-SUM-001 – Software User Manual
- [AD12] ASTOS-NEOPROP2-TRTP-001 – Test Report

2.2 Reference Documents

- [RD1] SPICE Tutorial,
ftp://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/22_fk.pdf
, Last accessed: 16-01-2013
- [RD2] The Asteroid Orbital Elements Database: <http://www.naic.edu/~nolan/astorb.html>,
Last accessed: 16-01-2013
- [RD3] Vallado D.A., Fundamentals of Astrodynamics and Applications, 3rd edition, 2008
- [RD4] TAI-UTC Conversion: Leap second table, <ftp://maia.usno.navy.mil/ser7/tai-utc.dat>, Last accessed: 16-01-2013
- [RD5] TT-UT1 Conversion: table, <ftp://maia.usno.navy.mil/ser7/deltat.data>
- [RD6] EOP: http://data.iers.org/products/39/640/orig/eopc04_05_IAU2000.62-now
- [RD7] EGM96 Model, The NASA GSFC and NIMA Joint Geopotential Model,
<http://cddis.gsfc.nasa.gov/926/egm96/egm96.html>, Last accessed: 16-01-2013
- [RD8] MPC, List of Observatory Codes,
<http://www.minorplanetcenter.net/iau/lists/ObsCodesF.html>, Last accessed: 16-01-2013
- [RD9] MPC, Format For Optical Astrometric Observations Of Comets, Minor Planets and Natural Satellites, <http://www.minorplanetcenter.net/iau/info/OpticalObs.html>,
Last accessed: 16-01-2013

- [RD10] MPC, Radar Astrometry Format,
<http://www.minorplanetcenter.net/iau/info/RadarObs.html>, Last accessed: 16-01-2013
- [RD11] MPC, Database Search, http://www.minorplanetcenter.net/db_search, Last accessed: 20-05-2015
- [RD12] Carpino M., Milani A. and Chesley S. R., Error statistics of asteroid optical astrometric observations, *Icarus* 166 (2003) 248–270, 2002
- [RD13] Guide to Minor Body Astrometry,
<http://www.minorplanetcenter.net/iau/info/Astrometry.html#cats>, Last accessed: 16-01-2013
- [RD14] TinyXml website, <http://www.grinninglizard.com/tinyxmldocs/index.html>, Last accessed: 16-01-2013
- [RD15] Włodarczyk I., Impact Orbits of the Asteroid 2009 FJ with the Earth, ISSN 0038-0946, *Solar System Research*, 2012
- [RD16] Time standards, <http://leapsecond.com/java/gpsclocck.htm>, Last accessed: 16-01-2013
- [RD17] NEO-SSA ESA program: <http://neo.ssa.esa.int/web/guest/search-for-objects>, Last accessed: 16-01-2013
- [RD18] Dallas S.S., Equations of Motion for Rotating Finite Bodies in the Extended PPN Formalism, 1976
- [RD19] Trenkel C., Kemble S. and Barraclough S., NEO Orbit Propagation including the Yarkowski Effect, EADS Astrium
- [RD20] Broucke R.A. and Cefola P.J., On the Equinoctial Orbit Elements, 1971
- [RD21] Milani A., Sansaturio M. E., Tommei G., Arratia O. and Chesley S.R., Multiple solutions for asteroid orbits: Computational procedure and applications, *Astronomy & Astrophysics* 431, 729-746, 2005
- [RD22] Vardaxis G. and Wie B., Impact Probability Analysis for Near-Earth Objects in Earth Resonant Orbits, AAS 14-427

3 Terms, definitions and abbreviated terms

3.1 Acronyms

The following abbreviations are used throughout this document.

Acronyms	
AD	Applicable Document
ADD	Architectural Design Document
CCD	Charge-coupled device
EOP	Earth Orientation Parameters
ESA	European Space Agency
GUI	Graphical User Interface
IAU	International Astronomical Union
ICD	Interface Control Document
jre	java runtime environment
LLS	Linear Least Squares
LOV	Line Of Variations
MC	Monte Carlo
MOID	Minimum Orbital Intersection Distance
NEA	Near Earth Asteroid
NEO	Near Earth Object
NEODyS	NEO Dynamic Site
PHA	Potentially Hazardous Asteroid
PPN	Parametrized Post Newtonian
RA	Right Ascension
RAAN	Right Ascension of the Ascending Node
RD	Reference Document
RMS	Root Mean Square
SoW	Statement of Work
SRD	Software Requirements Document
SRP	Solar Radiation Pressure
TAI	International Atomic Time
TDB	Time Dynamical Barycentric
TT	Terrestrial Time
VA	Virtual Asteroid

4 Conventions

Not Applicable

5 Purpose of the Software

In 2012 for the Space Situation Awareness program, Astos Solutions developed a new orbital propagator algorithm in order to assess the potential risk of impact of a NEO. It computes orbital properties, MOID, and close approaches of asteroids at Earth based on observation data provided by observatories and radar stations across the globe. In the continuation of the Orbit Propagation Algorithms for the Space Situational Awareness project, this new contract aims to improve the NEOPROP tool by implementing a list of identified missing features and requested improvements.

Once finalized, the updated software will be integrated into the ESA NEO Coordination Centre located in Frascati (Italy).

This document is the Software User Manual (SUM) prepared for the Project *Enhanced Orbit Propagator* performed by Astos Solutions under contract of the European Space Agency.

6 External view of the software

In this chapter an overview on the software structure (program files) and on the input/output files will be given. In the first paragraph the executable files will be briefly presented. Then the databases and libraries used will be shortly listed and introduced and finally the input and output files will be discussed.

In the main software folder four subfolders are contained:

- “bin” folder containing the executable files (see 6.1);
- “lib” folder containing all the input files to the models/libraries used by the software (see files described in 6.2);
- ‘doc’ folder containing the SUM and the two template configuration files, ‘template.conf’ (see 6.3.1.1) and ‘template.xml’ (see 6.3.1.3).

The other subfolders are introduced and discussed in 9.2.

6.1 Executable Files

According to the main architecture of the software, already described in [AD8] and [AD9], three executable files are provided:

- NEOPROP_MOID.exe;
- NEOPROP_prop_fast.exe;
- NEOPROP_prop_complete.exe.

Each executable is independent from the others. In chapter 9, each of them will be described.

Together with these three executables, a runnable jar file (Neoprop2.jar) is provided. A GUI has been developed to facilitate the test campaign and it has been included in the delivery, even though it was not requested in [AD2]. It has basic functionalities, which allow the user to run all the different modes of the software without using the command line. It allows to download ‘automatically’ observations (optical and radar) from internet, to get the NEO’s radius and absolute magnitude from NEODyS and to access online resources, relevant for the current test case. It allows partially to directly manipulate the input .conf and .xml files. The GUI has been used and tested only on Windows, therefore the compatibility with Linux is not guaranteed.

6.2 Libraries

NEOPROP2 makes use of some libraries and database. Some of them will be simply delivered with the rest of the code and do not require any external input file (e.g. TinyXml 6.2.5), while some others require external input files in order to initialize the model (e.g. Spice ephemeris 6.2.1). The user does not have to specify any specific settings or modify the input files, unless he wants to update/change the model in use.

All the external files used by these libraries are contained in the “lib” folder and are listed in the following table:

Table 1 List of files contained in the 'lib' folder

File Name	Info	Chapter	Used by
astorb.dat	Asteroid orbits database	6.2.2	A
cbm10.rules	Observatory statistical performance	6.2.6	A
cbm10.sta	Observatory statistical performance (depending on the star catalogue)	6.2.6	A
codes_300ast_20100725.bsp	Ephemeris kernel for Vesta, Hygea, Pallas and Ceres	6.2.1	N
codes_300ast_20100725.tf	Frame kernel for Vesta, Hygea, Pallas and Ceres	6.2.1	N
cspice.dll	C Spice code wrapper (Windows)	6.2.1	N
de430.lnx	Ephemeris kernel for planets (Linux)	6.2.1	A
de430.bsp	Ephemeris kernel for planets (machine-independent format)	6.2.1	N
egm96_100.txt	Earth spherical harmonics	6.2.7	N
eopc04.62-now	Earth orientation parameters	6.2.4	A
ET-UT.dat	ET - UT1 conversion table	6.2.3.2	A
leapSeconds.txt	TAI - UTC conversion table (leap s)	6.2.3.1	N
libcspice.so	C Spice code wrapper (Linux)	6.2.1	N
OBSCODE.dat	List of observatories codes	6.2.5	A
TAI-UTC.dat	TAI - UTC conversion table (leap s)	6.2.3.1	A

*In the 'Used by' column 'A' stands for Analytical module and 'N' stands for numerical module.

6.2.1 Spice Library

The Spice library [RD1] has been developed by NASA and it does not require any license in order to be used. The library has been compiled and integrated in the code. The 'cspice.dll' (for Windows) and 'libcspice.so' (for Linux) files are wrappers used by the numerical module to extract the ephemeris of a celestial body from kernels. The analytical module does not require such wrappers, since these computations are performed internally.

In order to compute ephemerides, input data are read from a file kernel, which has to be provided. Default kernels are included in NEOPROP2. The Numerical module now uses the "de430.bsp" kernel (updated version of the previously used de414), which contains the ephemerides of all the planets and of the Moon. This kernel has been chosen since it is the latest which contains ephemerides up to year 2201.

A second kernel (codes_300ast_20100725.bsp) is used in order to get ephemerides also for the main asteroid considered in the propagation: Vesta (Naif body name = '4 Vesta'), Hygea (Naif body name = '10 Hygea'), Pallas (Naif body name = '2 Pallas') and Ceres (Naif body name = '1 Ceres').

Moreover, a Frame Kernel can be imported along with the ephemeris kernel, since some ephemeris kernels are provided together with a specific frame kernel (the usual extension is ".tf"). The name and the location of the files should be the same, only the extension must be different (".bsp" for the kernel and ".tf" for the Frame kernel). Once an ephemeris kernel is used, NEOPROP2 searches for a frame kernel with the same name, and, if found, it is imported. The Numerical Module makes use of the codes_300ast_20100725.tf frame kernel.

In the Analytical Module the de430 kernel has been introduced too ("de430.lnx" file). In this module the spice ephemeris of asteroids are not used.

6.2.2 Asteroid Orbits Database

The Analytical module considers, during the trajectory propagation, gravitational perturbations generated by Vesta, Pallas and Ceres. The initial states and the physical parameters are taken from an external file: "astorb.dat". The format and the reference for this file can be found at [RD2]. Not the entire database is delivered with the NEOPROP source code, since only three asteroids are used. The database can be manually extended and additional asteroids could be used as gravitational perturbers as explained in 6.3.1.1.

6.2.3 Time Transformations Data

The time standards used and the name convention adopted in this software are widely used by the international community. A detailed description of time standards is outside the scope of this document. A detailed description can be found in [RD3].

6.2.3.1 TAI-UTC

To allow converting dates from TAI to UTC, the leap seconds need to be accounted. The table of leap seconds is stored in two dedicated files, located in the "lib" folder. This allows an easy maintenance of the tables. These files must be kept in the mentioned location.

The Analytical Module uses the "TAI-UTC.dat" file, while the Numerical Module uses the "leapSeconds.txt" file. The main difference between them is that this last file includes also the leap seconds before year 1972. The format is also slightly different.

The current values are taken from [RD4]. To update it, an additional row should be added at the end of the files. The last leap second included is from July 2015.

6.2.3.2 TT-UT1

The Analytical Module needs also to convert TT dates to UT1, since this transformation is used while processing the observation files. The table of values is stored in the "ET-

UT.dat" file. Also this file can be updated by adding a row at the end of the file. The current values are taken from [RD5].

6.2.4 Earth Orientation Data

The Earth Orientation Parameters describe the irregularities of the Earth's rotation. Technically, they are the parameters which provide the rotation of the terrestrial reference system to the celestial one as a function of time. This transformation is particularly important while processing the asteroid observations coming from ground stations.

The series of parameters used are from the EOP (IERS) 05 C04 model [RD6] and stored into the 'eopc04.62-now' file.

6.2.5 List of Observatory Codes

In order to be able to handle observations, the tool has to know the positions of the observatories used to make the observations. Therefore, a file containing a list of observatories with their positions is required: "OBSCODE.dat" (same format as in [RD8]).

6.2.6 List of Observatory Statistical Performance

Astrometric uncertainty is a crucial component of the asteroid orbit determination process. However, in the absence of rigorous uncertainty information, only very crude weighting schemes are available to the orbit computer. A method for carefully assessing the statistical performance of the various observatories that have produced asteroid astrometry, with the ultimate goal of using this statistical characterization to improve asteroid orbit determination is described in [RD12].

Therefore, a statistical RMS per observatory can be derived and used during orbit determination. Two input files are used to specify the RMS of any observatory using a CCD camera (their observations must have in the 15th column a "c", see 6.3.1.2).

The "cbm10.sta" file contains a list of observatory codes and for each of them two RMSs [""] values are provided (one for RA and one for declination measurements). This value is valid for any stars catalogue used for the data reduction [RD13].

The "cbm10.rules" file contains a list of observatory codes and different values of RMS [""] are provided depending on the stars catalogue used. The format used in this file is the following:

Table 2 Observatory Statistical Performance format

Characters 1-3	Character 4	Character 5	Characters 6	Characters 7-9	Characters 10-13	Characters 14-15	Characters 16-19
Observatory code	“.”	“c”	Catalogue code	“ @ ”	RMS for RA	“ , ”	RMS for declination

The possible catalogue codes are reported in the next table:

Table 3 Catalogue codes for the cbm10.rules file

Catalogue Code	Catalogue Name	Catalogue Code	Catalogue Name
a	USNO-A1.0	n	TRC
b	USNO-SA1.0	o	USNO-B1.0
c	USNO-A2.0	p	PPM
d	USNO-SA2.0	q	UCAC-2-beta
e	UCAC-1	r	UCAC-2
f	Tycho-1	s	USNO-B2.0
g	Tycho-2	t	UCAC-3-beta
h	GSC-1.0	u	UCAC-3
i	GSC-1.1	v	NOMAD
j	GSC-1.2	w	CMC
k	GSC-2.2	x	Hip-2
l	ACT	z	GSC (unspecified version)
m	GSC-ACT	' '	unknown
n	TRC		

For all the other observations (also not CCD) a default RMS is assumed:

- 3" if the observations was made before 1890;
- 2" if the observations was made before 1950;
- 1" for all the others.

6.2.7 Spherical Harmonics Gravity Field

A spherical harmonics gravity field is used by the Numerical Module during the trajectory propagation. The gravity fields of all planets and asteroids are assumed to be spherical, because they do not affect so much long term interplanetary trajectory propagation. On the other hand, the spherical harmonics of the Earth are considered, since they play an important role on the MOID computation. When the NEO's trajectory gets very close to the Earth, the asymmetries of the terrestrial gravity field can influence the fly-by.

An input file is used to initialize the spherical harmonics gravity field model ("egm96_100.txt"). The EGM96 model is currently used with 100x100 coefficients [RD7].

6.2.8 TinyXml Library

As it will be explained later in this document, the Numerical Module initializes the models and the propagation by using an input XML file. In order to read it and start all the computations a XML parser is required.

In NEOPROP the TinyXml parser is used [RD14]. It has a “zlib” license and the use for private or commercial purposes is free. This library will be not discussed or analyzed in this document, since all the documentation can be found in [RD14].

6.3 Input/Output Files

Several files are required by NEOPROP2 in order to start the computations. Moreover, new files are created as output by the Analytical and the Numerical module. Sometimes the output produced by the Analytical module is used as input for the Numerical module. Therefore, it is important to fix and explain the format of all these files so that the user will be able to create the input files and then to properly read the output.

These files will be presented and explained in the same order as they are requested and/or created by a complete run of NEOPROP2. First the input files for the Analytical module will be described and then their output, then the input files for the Numerical Module (full analysis) and finally its output files. The logic work-flow of the NEOPROP2 tool will be more detailed in chapter 9.

The output files are printed by the software after a successful run. They summarized the results of the computations and can be reused as input for a different task. All the output name files will have the same following structure:

IDspecificName.extension

The *ID* is a unique identifier (e.g. the name of the NEO) passed by the user to the tool (through command line see chapter 9) and then used for all the output files of the problem. It will allow the user to sort the files and to distinguish different test cases. Moreover, the input files of the Analytical module (observations, see 6.3.1.2) must have the same ID specified by the user.

The *specificName* identifies each file along with the *extension*. A unique combination of *specificName* and *extension* is assigned to each file.

For the output files printed by the Numerical module, a suffix will be added to the *specificName* in order to know if the Fast or the Complete analysis has been run (see [AD8] for more details). The suffix will be “_fast” or “_complete”, in this way it will be possible to distinguish the output.

Files are grouped in Input, Output and Input & Output subchapters, depending if they are used only as input or output files or if they are output of the Analytical module and then input to the numerical module.

6.3.1 Input

6.3.1.1 Configuration file

From this file some of the constants/settings used throughout the Analytical module are initialized.

The data is listed as a collection of tuples 'keyword = value notation'; each element is a keyword-value pair.

All the keywords (basically all what is before the "=" character) must not be modified, while the values can be customized by the user.

This file has seven different sections, with the following parameters:

- NEO's properties:
 - gravitational parameter of the NEO [km³/s²];
 - alternatively to the gravitational parameter, the mass of the NEO [kg] can be provided;
 - radius [m], if provided (directly or indirectly through the absolute magnitude) triggers the use of the solar radiation pressure in the equations of motion;
 - A1, A2, A3, deltaT and r0 parameters, related to the computation of the outgassing perturbation (as described in [AD6]), to be provided ONLY for comets.
- Perturbing planets: for each planet, the user should specify "T" if the gravitational perturbation of that planet should be considered or "F" if not;
- Perturbing asteroid: a list of perturbing asteroids can be listed here, for each of them the name, the ID (the same as in 6.2.2) and the gravitational parameter [km³/s²] has to be provided;
- Integrator: "ll" controls the step-size selection and the value should be between 6 and 13, if negative a fixed step-size integration will be used, "xl" is the fixed step-size [day] and "ml" is the maximum step-size [day];
- Gauss orbit determination: "minimum_rms" is minimum RMS that should be achieved before stopping the Gauss method, "is_neo_check" allows to turn off the check on the determined orbital elements (perihelion must be below 1.3 AU), "forward_loop" determines whether to run the "forward loop" ("T") and "backward_loop" (see paragraph 11.2.1.1 for a definition of these loops) allows to run the "backward loop" ("T"), "use_weighting" allows to use the new selection and grouping algorithms introduced in [AD6] and further explained in 11.2.1.1 (if true the backward and forward loops are not run);
- LLS orbit determination: "constrain_solution" determines if the analytical module should try to constrain the solution along the LOV in order to improve it. Depending on the dynamics of the problem, not always this technique is useful.
- Virtual Asteroids: "sigma" defines the maximum sigma deviation of a VA from the nominal solution (it can be set between 0.5 and 10), "number" defines the number of VAs (max 10,000), "second_eigen" determines if the second ("T") or the first ("F") eigenvalue will be used for the VAs generation and "scaling_lov" determines if the normalized LOV should be used to compute VAs.

Each section starts with the "##" characters. The order of the parameters within each section should not be altered.

Since some of the listed parameters might be test case specific, the user can specify different ".conf" files per each test case. In this case, this file should be placed in the "input" folder and follow the naming convention explained in 6.3 (*specificName* = ""; *extension* = ".conf"). If a local file is not found, the tool will use the "template.conf" file present in the "doc" folder.

The structure of the configuration file is summarized in the following table:

Table 4 Configuration file input summary

Property name	Unit	Optional	Default	Note
neo.mass	kg	X	-	mass or mu must be provided.
neo.mu	kg ³ /s ²	X	-	mass or mu must be provided.
neo.albedo	-	X	0.154	It can be used to estimate the radius together with the magnitude.
neo.magnitude	-	X	-	It can be used to estimate the radius together with the albedo.
neo.radius	m	X	-	If radius or magnitude is provided the solar radiation pressure is computed.
neo.A1	AU/day ²	X	-	If provided, the outgassing perturbation is computed. It is the radial non-gravitational acceleration parameter.
neo.A2	AU/day ²	X	-	If provided, the outgassing perturbation is computed. It is the transverse non-gravitational acceleration parameter.
neo.A3	AU/day ²	X	-	If provided, the outgassing perturbation is computed. It is the normal non-gravitational acceleration parameter.
neo.deltaT	day	X	0.0	Asymmetric outgassing perturbation w.r.t the perihelion.
neo.r0	AU	X	2.808	Distance beyond which re-radiation of solar energy begins to dominate the use of this energy for vaporizing the comet's ice.
asteroid	-	X	-	If present, name and mu must be provided too.
asteroid.name	-		-	
asteroid.mu	kg ³ /s ²		-	
integrator.radau.ll	-		-11	
integrator.radau.xl	day		1	
integrator.radau.ml	day		0.1	
gauss.is_neo_check	-	X	true	When false, it allows to analyse any celestial body.

gauss.minimum_rms	-	X	0.001	normalized rms
gauss.forward_loop	-	X	false	
gauss.backward_loop	-	X	false	
gauss.use_weighting	-	X	true	
lls.constrain_solution	-	X	false	
vas.sigma	-		4	
vas.number	-		100	
vas.second_eigen	-		false	
vas.scaling_lov	-	X	true	

6.3.1.2 Observations

Optical and radiometric observations are required as input for the Analytical Module in order to run the orbit determination algorithm. These are the only input files required by this module.

Two types of observations can be provided: optical and/or radiometric. Both these two types can be provided at the same time for a certain test case, but in two separate files having different format.

- Optical observations (*specificName* = ""; *extension* = "obs")

The format used is shown here:

Table 5 Optical Observations Format [RD9]

Characters	Format	Use
1-14	A14	Name (not used)
15	A1	Type of Observation
16-32	A17	Date of observation
33-44	A12	Observed RA (J2000.0)
45-56	A12	Observed Declination (J2000.0)
57-65	9x	Blank
66-71	F5.2,A1	Observed magnitude and band
72-77	x	Blank
78-80	A3	Observatory Code

- Radiometric observations (*specificName* = ""; *extension* = "rad")

The proper format to be used is shown here:

Table 6 Radiometric Observations Format

Characters	Format	Use
1-41	A41	Name (not used)
42-60	A17	Date of observation
61-74	A14	Measurement: Range [km] or Velocity [km/day]
75-76	2x	Blank
77-82	A6	A priori rms
83	x	Blank
84-85	A2	Type measurement: R for Range or V for Doppler
86-91	A6	Blank
92-95	A4	Frequency
96-99	4x	Blank
100-101	A2	Observatory Code Transmitter
102-104	3x	Blank
105-106	A2	Observatory Code Receiver

For both optical and radiometric observation files, some examples are delivered along with the rest of the code.

The user must remember to leave 1 empty line at the end of each file.

6.3.1.3 XML file

The Numerical module allows the user to customize some of the models used (e.g. integrator, perturbations, etc). All these settings are collected in a XML file that will be read by the executable in order to process the data and to perform all the computations. **The name of the tags in the XML file should NEVER be touched by the user.** Only the contents might be changed. Therefore, the list of possible inputs which can be specified in the XML file will be listed in chapter 11, where all the models will be discussed.

The user can specify a different xml file per each test case. In this case, this file should be placed in the "input" folder and follow the naming convention explained in 6.3 (*specificName* = ""; *extension* = ".xml"). A 'template' file (template.xml), to be used as starting base for a new test case, can be found in the "doc" folder.

6.3.1.4 GUI Configuration file

This file (*specificName* = ""; *extension* = ".neoprop") stores all settings showed by the GUI in the configuration panel (11.2.4.1). It must be located at the same level as the input and output folders (introduced in 9.3). The file is updated every time the File->Save button is

pressed or when a new test case is created. All the information saved in the file is listed below line by line:

Table 7 GUI Configuration file format

GUI Setting Label	Setting Info	Optional	Type
Test case folder	Absolute path of the latest loaded test case		String
NEO ID	ID of the NEO to analyse, it is used to determine all file names and retrieve observations and reference values from internet.		String
Analytical Module	Defines if the analytical module has to be run.		Boolean
Final Epoch JD (TDB)	If it is different from 0.0, the determined state is propagated till this epoch.	X	Double
Compute VAs	Defines if to compute VAs.	X	Boolean
Log file	Text file name used to save the output stream of the analytical module.	X	String
Clean output folder	If it is selected, all the output files generated by the Analytical module are deleted before running it, preventing it from using the old '.ci' file as preliminary solution for the LLS method (see 11.2.1.1).	X	Boolean
Numerical Module	Defines if the numerical module has to be run.		Boolean
Final Epoch (TDB)	Defines the final epoch of the trajectory propagation performed by the numerical module.	X	3 Integers
Run 'Fast' Mode	Defines if the 'fast' numerical module has to be run.	X	Boolean
Log file	Text file name used to save the output stream of the 'fast' numerical module.	X	String
Run 'Complete' Mode	Defines if the 'complete' numerical module has to be run.	X	Boolean
Log file	Text file name used to save the output stream of the 'complete' numerical module.	X	String

If the 'Analytical Module' or 'Numerical Module' flag is set to false, the module-specific settings are not loaded/stored in the file, for this reason they are marked as optional.

6.3.1.5 GUI Properties file

This file contains some basic properties, loaded by the GUI every time it is run. The file name is always 'properties.dat' and it must be located in the 'bin' folder.

The first line contains the GUI width and height (in pixels), separated by an empty space. From the second line on, the absolute path of the last 4 test cases is stored.

The file is overwritten when GUI is closed, in order to update these properties.

6.3.2 Input & Output

6.3.2.1 Determined States, Covariance Matrix, Analytical MOID and Used Observations Files

In this paragraph, all the files produced by the Analytical Module will be described. Two files are created every time this module is run successfully:

- Determined States and Analytical MOID (specificName = ""; extension = "ci")

In this file the determined states, covariance matrix and Analytical MOID are printed along with some other parameter. The format of the file is shown here:

Table 8 Determined States and Analytical MOID file format

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
ID	-	-	-	-	-
Julian Date TT	-	-	-	-	-
SMA [AU]	Eccentricity [-]	Inclination [°]	RAAN [°]	Argument periapsis [°]	Mean Anomaly [°]
SMA-1 σ [AU]	Eccentricity-1 σ [-]	Inclination-1 σ [°]	RAAN-1 σ [°]	Argument periapsis-1 σ [°]	Mean Anomaly-1 σ [°]
MOID [AU]					

After these 5 rows, the 6x6 covariance matrix is printed if successfully computed. The units used are the same as reported in Table 8.

This file is used also as input for the Numerical Module and it is compatible with the previous version of the tool. Even though it contains almost the same information stored in the '.ele' file, it allows to provide additional parameters (e.g. MOID) which are not part of the '.ele' and '.equ' files but are required by NEOPROP2. On the other hand, the format of '.ele' and '.equ' files is in line with the one used by NEODYs.

- Determined Keplerian elements (specificName = ""; extension = "ele")

In this file the determined Keplerian elements, RMSs and covariance matrix are printed along with some other parameter. The format of the file is shown here:

Table 9 Determined Keplerian elements file format

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
----------	----------	----------	----------	----------	----------

ID	-	-	-	-	-
Julian Date TDB	-	-	-	-	-
SMA [AU]	Eccentricity [-]	Inclination [°]	RAAN [°]	Argument periapsis [°]	Mean Anomaly [°]
SMA-1 σ [AU]	Eccentricity-1 σ [-]	Inclination-1 σ [°]	RAAN-1 σ [°]	Argument periapsis-1 σ [°]	Mean Anomaly-1 σ [°]

After these 4 rows, the 6x6 covariance, normal and correlation matrices are printed if successfully computed. The units used are the same as reported in Table 9. Before column one, a string is printed in order to help the user to identify the provided parameters in the same way as done by NEODYs in 'equ' files.

- Determined equinoctial elements, weak direction vector, RMSs and covariance matrix (specificName = ""; extension = "equ")

In this file the determined states, covariance matrix and Analytical MOID are printed along with some other parameter. The format of the file is shown here:

Table 10 Determined equinoctial elements file format

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
ID	-	-	-	-	-
Julian Date TDB	-	-	-	-	-
SMA (a) [AU]	h [-]	k [-]	p [-]	q [-]	λ [°]
SMA-1 σ [AU]	h -1 σ [-]	k-1 σ [-]	p-1 σ [-]	q-1 σ [-]	λ -1 σ [°]

After these 4 rows, if the virtual asteroids have been computed the eigenvector components identifying the LOV are printed (preceded by the keyword '! WEA '). Then the 6x6 covariance, normal and correlation matrices are printed if successfully computed. The units used are the same as reported in Table 10. Before column one, a string is printed in order to help the user to identify the provided parameters in the same way as done by NEODYs in 'equ' files.

- Used observations (specificName = ""; extension = "data")

In this case, the header file describes the content. It is worth noticing that this file contains the full list of observations along with two new parameters:

- Khi: is the residual of the observation;
- Used flag: the last column has a 1 for each observation used during the orbit determination and a 0 for each observation discarded.

Moreover, the second line of the file contains the total RMS of the residual in arcsec.

6.3.2.2 VAs on LOV

If requested by the user, a list of virtual asteroid can be computed along the LOV. In this case an additional file is printed (*specificName* = ""; *extension* = "va").

For each VA computed a new row is created with the following parameters:

Table 11 VAs on LOV file format

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
SMA [AU]	Eccentricity [-]	Inclination [°]	RAAN [°]	Argument perigee [°]	Mean Anomaly [°]	σ [-]

The σ value of a VA is the position of an orbit along the LOV in sigma space. The farther it is from zero (nominal orbit), the less likely the orbit. Plausible orbits usually fall into the interval [-3, 3] or [-4, 4] [RD15].

6.3.3 Output

6.3.3.1 Propagated Nominal Trajectory

This file (*specificName* = "_nominal_trajectory"; *extension* = "txt") is outputted every time the Numerical module is run and output interval selected in the XML file is not zero.

In this file all the states and derivatives are reported at each output time. Then also the polar coordinates, the Cartesian relative position vector (position vector in the PCPF reference frame, see Appendix A – Reference Frames) and the orbital parameters of the trajectory are always reported.

All the output (except for the Cartesian relative position vector) is expressed always in the reference frame used to integrate the trajectory (Inertial J2000).

The trajectory file reports in the first line (header) all the variables with relative units printed in the file. For this reason the full explanation of the trajectory file format will not be assessed here; basically the trajectory file contains all the requested information to be properly read and understood.

6.3.3.2 Third Bodies Relative Positions

This output file collects the evolution of the position vectors of the NEO w.r.t. all the celestial bodies defined (except for the Sun). If only the central body is specified (inactive third body perturbation), then this file is not created.

The format *specificName* = "nominal_thirdBodiesRelativePos"; *extension* = "txt") of this file is here summarized:

Table 12: Third Bodies relative Positions output file format

Column 1	Column 2	Column 3	Column 4-6	Column 7	Column 8-10
Simulation time [s]	Julian Date [day]	Calendar Date [-]	X-Y-Z Inertial Position [m]	Distance [km]	X-Y-Z Relative Position [m]

For each celestial body the columns 4-10 are provided. The inertial position vector should be considered given in the same reference frame used for the integration (Inertial J2000) and outwards from the celestial body towards the asteroid, while the relative position is provided in the rotating PCPF reference frame.

This file is created only if the "Print_Third_Body_Trajectories" flag is specified in the XML file (see 11.2.2.2.1).

6.3.3.3 Close Approaches and Numerical MOID

This file (*specificName* = "_close_approaches"; *extension* = "txt") is outputted every time the Numerical module is successfully run.

The first line specifies the timeframe of the numerical analysis (initial and final epoch of the numerical integration), while the second line reports the minimum close approach distance which has been considered to record a close approach. Only close approaches with a distance from a celestial body below the one specified by the 'Minimum_Close_Approach_Distance_To_Record' parameter (defined in the XML and equal to 0.005 AU by default, see 11.2.2.2.1) are reported.

Then follows a list of all the nominal close encounters to any planet the NEO had during the trajectory propagation. In case of a complete analysis, only the nominal trajectory is considered. The epoch and the distance are always printed.

Then, in case of a complete analysis, Earth virtual impactors (minimum distance below 1.5 * Earth radius) are printed and for each of them the impact risk is assessed by providing the asteroid mass, kinetic energy, impact probability, Torino and Palermo scale values.

6.3.3.4 VAs Trajectory Data

This file (*specificName* = "_VAs_final"; *extension* = "txt") is outputted every time the "Complete analysis" is successfully run by the Numerical module. For each VA provided to the tool and propagated, the following properties are reported:

Table 13 VAs Trajectory Data file format

Printed only if the VA has a close approach to the Earth			
Columns 1-6	Columns 7-9	Column 10	Column 11
*Orbital Elements of the	Geocentric Cartesian	Minimum distance of	Epoch of the Earth

final orbit	states at the Earth closest approach [m]	the Earth closest approach [AU]	closest approach (TT)
-------------	---------------------------------------------	------------------------------------	-----------------------

*same units and order as in Table 8 (3rd row)

6.3.3.5 Error Log File

While running the Numerical module, if the program detects some error (in the initialization, computations, models, etc), the simulation is stopped and an error log file is printed. In this file the location (file and line) and an error message for each error detected is added. This file should be used by the developers to fix the problem.

At the same time an error message is printed on the screen in order to help the user to solve the problem.

If an error occurs during the execution of the Analytical module, the program is stopped and an error message is printed on the screen.

7 Operations environment

7.1 General

This software has been developed in order to be used on a PC and independent from third party software.

7.2 Hardware configuration

The minimum hardware requirements are 2GB RAM and 1GB of free hard disk memory.

Some parts of both the modules have been coded in order to allow parallelization. Since some computations might be really "expensive" in terms of time and memory consumption, the use of a PC with a multi core (at least 4) processor and with at least 4 GB of ram is strongly recommended.

7.3 Software configuration

The operating system should be Windows 7 or superior.

7.4 Operational constraints

Not applicable.

8 Operations basics

Not applicable.

9 Operations manual

9.1 General

In this chapter the main operations to successfully install and use NEOPROP2 are discussed.

9.2 Set-up and initialization

The NEOPROP2 tool is delivered with the follow subfolders:

- “bin” containing the executables (see 6.1) and the compiler libraries required at runtime:
 - under Windows (compiler MinGW 4.6.2): libgcc_s_dw2-1.dll, libgfortran-3.dll, libgomp-1.dll, libpthread.a, libpthread-2.dll, libquadmath-0.dll and pthreadGC2.dll;
 - under Linux (compiler MinGW 4.9.1): linux-gate.so.1, libgomp.so.1, libpthread.so.0, libgfortran.so.3, libm.so.6, libgcc_s.so.1, libquadmath.so.0, libc.so.6, libdl.so.2, libstdc++.so.6
- “lib” containing all the libraries and the input files for the NEOPROP2 models which are not dependent on the specific test case (see 6.2);
- “doc” containing the SUM, the .conf and .xml template files and a text file reporting the sources where the GUI icons have been taken from;
- “src” containing the source code;
- “jre” containing a local copy of a Java installation (required at least jre 8 update 45);
- “Test_cases” containing the delivered test cases.

The main NEOPROP2 folder contains also a batch file (“Start_NEOPROP.bat”), that can guide the user through the different modes, as alternative to the GUI, and another batch file (“Neoprop2_local_jre.bat”) able to start the GUI using the local Java installation contained in the “jre” folder, in case Java is not installed on the computer.

9.3 Getting started

In order to run the tool a test case folder has to be created (e.g. in the “Test_cases” folder). It must have two subfolders: *input* and *output*. The input folder hosts all the observation files, the configuration and the xml file, while the output folder hosts all the output files printed by the tool (determined states, propagated trajectory, etc.).

Depending on the executable the user is going to run, a different set of files must be present in the *input* folder. The next table summarizes the list of files required:

Table 14 Test case “input” folder structure for each executable

Executable	List of input files
NEOPROP_MOID.exe	Observations: “ID.obs” (and, if available, “ID.rad”)

	Configuration file: "ID.conf"
NEOPROP_prop_fast.exe	Determined states: "ID.ci" XML file: "ID.xml"
NEOPROP_prop_complete.exe	Determined states: "ID.ci" XML file: "ID.xml" VAs file: "ID.va"

The path of the test case folder and the ID to be used must be specified while invoking the executable. The list of arguments for each executable will be specified in the next paragraph.

Once a new test case folder has been created or an old one has been chosen, the user can start using the software. In case the user wants to rerun an old test, he should be aware that ALL the output files will be overwritten.

9.4 Normal operations

The only operation available to the user is to run one of the executables. Depending on the executable, a different type of analysis is run:

Table 15 Analysis run by each executable

Executable	Analysis/Task
NEOPROP_MOID.exe	Orbit determination Computation of the analytical MOID VAs computation
NEOPROP_prop_fast.exe	Trajectory propagation of the initial state Computation of the numerical MOID and close approaches
NEOPROP_prop_complete.exe	Trajectory propagation of the initial state Computation of the numerical MOID and close approaches Trajectory propagation of VAs Impact risk assessment

More details about each type of analysis will be provided in chapter 11.

While invoking one of the NEOPROP2 executables, the user must specify a short list of arguments in order to provide some important information, e.g. location of the test case folder. Some of the arguments are "optional". Even though they do not need to be specified, it is always recommended to specify all of them.

The list of arguments for each executable is specified here:

Table 16 List of arguments for each executable

Executable	List of arguments
NEOPROP_MOID.exe	Executable absolute path Test case folder absolute path ID (e.g. NEO's name) VAs computation flag ("true" or "false")* Final propagation time** Mono-core flag ("true" or "false")***
NEOPROP_prop_fast.exe	Executable absolute path Test case folder absolute path ID (e.g. NEO's name)
NEOPROP_prop_complete.exe	Executable absolute path Test case folder absolute path ID (e.g. NEO's name)

*this flag has to be provided as string, if not specified it is assumed to be "false" by default. If "true" is specified, the computation of VAs along the LOV will be run.

**this parameter can be specified only if the VAs computation flag is specified. It can be used to propagate the determined state to a certain epoch (TDB). This feature is particularly useful for testing, but should be set to 0.0 in case the VAs computation flag is set to "true". In this case, the epoch is the one corresponding to the second observation used by the Gauss method to determine the preliminary solution. This is the standard approach to get the best RMS.

***this parameter can be specified only if the previous two optional parameters are specified. If "true" only 1 core is used, if "false" (as by default), all the available cores will be used.

9.5 Normal termination

When the execution of a task (executable) is completed successfully, the following message is printed by the software: "Computation completed".

9.6 Batch File

As mentioned in 9.2, a batch file is delivered together with the executables and the source code. The purpose of this file is to guide the user through the different operational modes and to show how the different tasks (explained in 11.1) can be run together in order to perform a complete analysis of a NEO.

Before running the "Start_NEOPROP.bat" file, the user should open it and edit two variables defined at the beginning of the file:

- TEST_CASE_FOLDER is the absolute path of the test case folder. The default path is the Apophis test case which is provided along with the tool. The user should specify the folder of the test case he wants to run.
- NAME_ASTEROID is the Identifier (ID) of the NEO the user wants to analyse. This ID is used in order to run the proper input files from the test case folder.

These two parameters have been already introduced and explained in 9.3 and 9.4. Once they have been correctly assigned, the batch file will allow to run one of the following task:

1. Analytical Module (11.2.1): only the NEOPROP_MOID.exe executable will be run;
2. Fast Numerical Module (11.2.2): only the NEOPROP_prop_fast.exe executable will be run
3. Complete Numerical Module (11.2.3): only the NEOPROP_prop_complete.exe executable will be run;
4. Full Analysis: the NEOPROP_MOID.exe executable will be run and the output printed will be used as input for the execution of NEOPROP_prop_complete.exe.

The batch file shows to the user also how to properly invoke the executables. This is the first step of an automatic system which will be able to monitor all NEOs.

9.7 Error conditions

See chapter 10.5.

9.8 Recover runs

Not applicable.

10 Reference manual

10.1 Introduction

NEOPROP2 tries to help the user in order to be properly set and used. The helps used are error/warning messages before running any task (during initialization) and during the computation.

10.2 Help method

The main help for any user is always the SUM.

10.3 Screen definitions and operations

Not applicable.

10.4 Commands and operations

Not applicable.

10.5 Error messages

Two types of error/warning messages are provided to the user:

- Input checks
- Computation checks.

Error messages allow the user to select the proper set of input in order to successfully run the selected task. For instance, if the user does not provide the proper set of input files, an error message will be printed informing the user.

The NEOPROP2 cores check also if all the inputs required are present (e.g. in the XML file). If not the computation is stopped and a message is printed. For instance if the user activates the third Relativistic Effects perturbation but forgets to insert the celestial bodies to be considered, an error message will appear. Indeed, these messages try to prevent wrong uses of the NEOPROP2 functionalities, which might lead to undesired crashes or wrong results.

Moreover, error messages or warning can be provided also during the computation itself. For instance if the integrator reaches the minimum step-size or if the S/C reaches a negative value of the altitude over the central body, the process is stopped and an error message is printed in the GUI output log. Whenever a task is run and an error occurs, the computation is always stopped, an error message is displayed and an error file is printed (as described in 6.3.3.5).

11 Tutorial

11.1 Introduction

The NEOPROP2 software has two separate modules:

1. The Analytical Module makes use of analytical algorithms in order to rapidly assess the impact risk of a NEO. It is responsible for the preliminary analysis. An Orbit Determination algorithm will determine the initial state, along with its uncertainty, and the MOID of the NEO (analytically). If the MOID will be under a certain threshold a numerical analysis will be recommended.
2. The Numerical Module makes use of numerical algorithms in order to refine and to better assess the impact probabilities of a NEO (detailed analysis). The initial state provided by the Orbit Determination process (of the Analytical Module) will be used to numerically propagate the trajectory. The numerical propagation can be run in two modes:
 - one faster, in order to get a fast evaluation of the trajectory (fast analysis). In this case, only a fix list of basic perturbation models will be considered.
 - one more precise taking into consideration more detailed models (complete analysis). Along with the numerical propagation of the nominal state, several VAs will be numerically propagated in order to determine the MOID. This MOID computation differs from the analytical one since it takes into consideration the full dynamics of the problem (all the relevant perturbations). Moreover, in case at least one virtual impactor is found, the impact risk is assessed.

11.2 Using the software on a typical task

All the different tasks of the software previously listed will be analysed separately here. In principle each function is independent from the others and can be run alone (if all the input files required and listed in 9.3 are available). A step-by-step procedure is provided in each section to help the user getting to know the software.

11.2.1 Analytical Module

11.2.1.1 Introduction

This module allows determining the initial state (with uncertainty) of a NEO from a set of observations. First a preliminary Gauss algorithm is run in order to get a first solution of the orbit determination problem. For this solution only optical observations are used. If in the test case output folder a “.data” file (see 6.3.1.2) and a “.ci” file (see 6.3.2.1) are found, then the preliminary solution is not recomputed, but it is imported from the “.ci” file.

The Gauss method needs only three observations in order to compute a preliminary initial state. The first, last and intermediate observations are by default the first ones to be used. Then, if the new weighting algorithm is not selected, the tool tests the following:

- 1) the function checks which intermediate observation gives the best solution (smaller RMS) if used along with the first and last observations;

- 2) if requested by the user a "forward loop" is started: the tool checks which 2 observations give the best solution if used along with the last observation;
- 3) if requested by the user a "backward loop" is started: the tool checks which 2 observations give the best solution if used along with the first observation;

If the new weighting algorithm is requested, all triplets matching the following criteria are tested:

- the time difference between two consecutive observations must be at least 0.03 days;
- the time difference between two consecutive observations must be maximum 150 days;
- the weight (computed considering an ideal time difference between two consecutive observations equal to 20 days) assigned to a triplet must be lower than the weight assigned to the previously tested triplets;
- the statistical RMS assigned to the observatory related to each observation must be lower than 4.0 arcsec.

As many threads as possible are used to evaluate the triplets. The search is stopped once the specified minimum RMS is reached. If not reached, the triplet with the lowest RMS is used. Then the preliminary solution is improved by mean of a LLS technique, which eventually considers also radiometric observations.

Alternatively, if a ".data" file and a ".ci" file are found, the old observations are imported and the precomputed initial state is used as preliminary solution. In this way only new observations are considered by the LLS technique, without having to "refit" the same observations already considered. The user should check that the same observations are not present in both the ".data" file and in the new observation files (".rad" and ".obs"), otherwise they will be considered twice.

If the LLS technique does not converge, the Gauss method is rerun and a new preliminary solution is found (without using observations used for previous preliminary solutions). After 30 unsuccessful iterations the software stops, saying that the maximum number of iterations has been reached without obtaining convergence. This can happen if observations are not good enough, or if an abrupt event (e.g. close approach) modified the orbit within the observational arc.

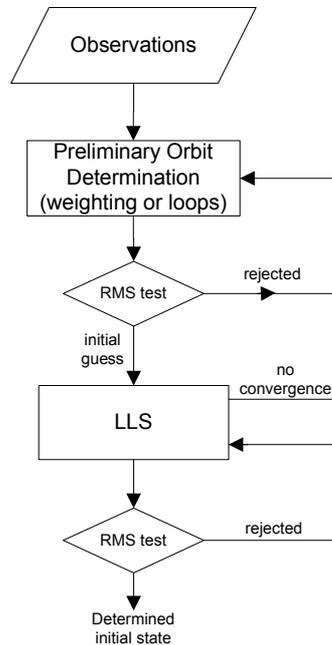


Figure 1 Orbit determination algorithm diagram

If requested by the user (see 9.4), the initial states of a set of VAs are computed. The vector, identifying the LOV, is computed from the covariance matrix, corresponding to the LLS solution, by using the first or the second eigenvalue (depending on the user settings). Each VA is computed from the previous one by moving along the LOV. An iterative technique is used to assure that the new VA belongs to the LOV, whose direction vector is indeed recomputed constantly. The user can decide if the LOV should be computed with or without scaling, as explained in [RD21].

Finally, the determined initial state of the NEO is used in order to analytically compute its MOID by using the Sitarski method. All these orbit determination techniques are explained in [AD3], [AD4] and [AD5].

11.2.1.2 Input

The inputs required to run the Analytical module are of two different types and have been already assessed in this document and are summarized here:

- the list of arguments to invoke the executable “NEOPROP_MOID.exe” (see 9.4);
- a file with optical observations and, optionally, a file with radiometric observations (see 6.3.1.2 for the file formats and 9.2 for the file name convention).

11.2.1.3 Step-by-Step Procedure

In case the user needs to create from scratch a new test case, a valid test case folder structure should be generated as presented and explained in section 9.2. The 2013 TP4 NEO will be used for this example. The ID mentioned in chapter 6.3, which is used to generate input and output files with the proper naming convention, can be set to

2013_TP4, even though any other ID would work too. All computations do not take into consideration the selected ID, but just the data stored in the files.

The following steps are required to generate a valid test case:

1. Create a new folder, which will contain all the input/output files. For this example the folder can be named 'Test'.
2. Create inside the 'Test' folder two subfolders: 'input' and 'output'.
3. Place inside the 'input' folder the necessary two input files:
 - a. **2013_TP4.obs**, which contains the optical observations of the object according to file format described in paragraph 6.3.1.2. This data can be obtained from the following webpage: http://www.minorplanetcenter.net/tmp/2013_TP4.txt. If the user has radiometric observations, the *2013_TP4.rad* file should be created according to the file format described at 6.3.1.2.
 - b. **2013_TP4.conf**, which contains the input parameters for the analytical module. The 'template.conf' file, present in the 'doc' folder, can be used as starting point for the creation of the aforementioned file. A copy of this file should be created and placed into the 'input' folder and then renamed into *2013_TP4.conf*. Then it should be edited in order to use the correct settings for the current object, especially physical parameters as mass (or the gravity parameter μ) and the radius (or the magnitude). For this test, the following modifications must be applied:

- 'neo.mu= 6.5e-10' should be replaced by 'neo.mass= 2192127.608921'
- 'neo.radius= 200.0' should be replaced by 'neo.magnitude= 27.304'

The meaning of all the other parameters can be found in section 6.3.1.1.

Alternatively to the procedure just described, the user could make use of the GUI to generate a new test case, including all the requested input files, as explained in section 11.2.4.2). Furthermore, The GUI could be used to execute the test case, as showed in paragraph 11.2.4.4.

If the user wants to run the test case from command-line, the "NEOPROP_MOID.exe" executable should be invoked with the proper set of arguments as described in Table 16. To open the command prompt window, run cmd.exe on Windows or use the keystroke Ctrl+Alt+t to open the terminal for Linux. The command-line call is showed below:

```
"NEOPROP_installation_folder_absolute_path\bin\NEOPROP_MOID.exe"  
"Test_folder_absolute_path\Test" "2013_TP4" "false" "0.0" "false"
```

A similar call is required to run the delivered 'Test_Case_0020', present in the 'Test_cases' folder. The following call must be used:

```
"NEOPROP_installation_folder_absolute_path\bin\NEOPROP_MOID.exe"  
"NEOPROP_installation_folder_absolute_path\Test_cases\Test_Case_0020" "99942"  
"false" "2456400.5" "false"
```

It is worth noticing that the first two arguments are the absolute paths of the executable and of the newly created 'Test' folder. The "0.0" argument stands for the final propagation epoch, expressed as Julian date TDB. A value different from 0 allows the user to propagate the determined nominal state till the provided epoch. This allows a fast comparison between states generated by different tools. The default value 0.0 tells the tool not to propagate the computed nominal state of the NEO. This is a precondition to compute virtual asteroids, because the propagation of state and covariance matrix with the analytical module would cause a much larger error than the one obtained by creating

virtual asteroids from the nominal state and propagating them with the numerical module. In this case, the epoch is the one corresponding to the second observation used by the Gauss method to determine the preliminary solution. This is the standard approach to get the best RMS.

At the end of the execution the following files (see 6.3.2.1 for the files formats) are generated in the 'output' folder:

- 2013_TP4.ci
- 2013_TP4.ele
- 2013_TP4.equ
- 2013_TP4.data

In case the user wants to create virtual asteroids the following steps are required:

- The first Boolean (in the list of arguments for the command-line) should be set to 'true'.
- At the end of the 2013_TP4.conf file, the number of virtual asteroids to be created can be modified (by default it is set to 5000).
- **The output folder should be 'cleaned'** (as done by the GUI when the corresponding checkbox is selected, see also Table 7). If the analytical module finds the 2013_TP4.ci and the 2013_TP4.data files, it tries to import from them the initial state (as explained in [AD12]) and it tries to fit the observations found in the 2013_TP4.obs, which are assumed to be new. This feature should be used only when new observations are actually available and allows the user to fit only them starting from the previously computed initial state without the need to restart the all process.

The new command-line call for the generation of virtual asteroids is showed:

```
"NEOPROP_installation_folder_absolute_path\bin\NEOPROP_MOID.exe"  
"Test_folder_absolute_path\Test" "2013_TP4" "true" "0.0" "false"
```

At the end of the execution the following file is created:

- 2013_TP4.va

11.2.1.4 Output

The outputs of this task are text files:

- ".ci" file with the determined initial state, uncertainties and MOID (see 6.3.2.1);
- ".ele" and ".equ" files reporting additional information about the determined initial state (see 6.3.2.1);
- ".data" file with the used observations and residuals (see 6.3.2.1);
- ".va" file with the determined initial states of VAs (see 6.3.2.2). This file is printed only if the argument "VAs computation flag" (9.4) was set to "true".

11.2.2 Numerical Module (fast analysis)

11.2.2.1 Introduction

The Numerical module performs numerical integrations. Its “fast analysis” is supposed to preliminarily assess the trajectory of a NEO. When the MOID computed by the Analytical module is small enough to be potentially dangerous for the Earth (e.g. smaller than 0.005 AU), a numerical integration is recommended. During this fast analysis, the initial state provided is numerically integrated and the MOID computed.

Even though the propagator used here shares the same models and XML input file with the one used in the “complete analysis”, it does not allow the same flexibility as the other does. For instance, a fixed set of perturbations is used during this analysis, while in the “complete analysis” the user can freely customize them. This approach has been chosen for two reasons:

1. the fast analysis is supposed to be “fast”. For this reason only some basic (and most important) perturbations are considered;
2. the fast analysis allows to be run also by users not so familiar with numerical propagations. For this reason many settings are initialized by default and the user will not have to care for them.

11.2.2.2 Input

The inputs required to run this module are of two different types:

- the list of arguments to invoke the executable “NEOPROP_prop_fast.exe” (see 9.4);
- an XML file with the settings required to initialize the propagator (see 6.3.1.3 and 9.2 for the file name convention);
- a ‘.ci’ file with the initial state of the NEO (see 6.3.2.1 for the format).

The XML file can be shared by the fast and the complete analysis. Since the fast analysis initializes some models by default, only a certain part of the XML file is parsed and indeed used. This part will be described here, while those parts read only during the complete analysis will be addressed in 11.2.3.2.

The main node tag of the XML file is “NEOPROP”. Inside this node there:

- “Propagation”
- “Asteroid”
- “Virtual Asteroids” (not used for this task, but described in 11.2.3.2.3)

Each of them represents a different class and will be described in the next paragraphs.

11.2.2.2.1 Propagation

The propagation node is reported here:

```

<Propagation>
  <Initial_Conditions>
    <Input_File>W:\Eclipse workspace\NEOPROP2\Test cases\Apophis\output\99942.ci</Input_File>
  </Initial_Conditions>
  <Integration>
    <Integrator>Gauss_Jackson_8_Adapted</Integrator>
    <Minimum_Step_Size>0.01</Minimum_Step_Size>
    <Maximum_Step_Size>8640.0</Maximum_Step_Size>
    <Local_Tolerance>1.0E-13</Local_Tolerance>
    <Output_Interval>86400.0</Output_Interval>
    <Final_Epoch>
      <Calendar_Date>
        <Year>2029</Year>
        <Month>4</Month>
        <Day>17</Day>
        <Hour>0</Hour>
        <Minute>0</Minute>
        <Second>0.0</Second>
      </Calendar_Date>
      <Time_Standard>UTC</Time_Standard>
    </Final_Epoch>
  </Integration>
  <Perturbations>
    <Third_Body_Perturbations>
      <Active>true</Active>
    </Third_Body_Perturbations>
    <Solar_Radiation_Pressure>
      <Active>true</Active>
    </Solar_Radiation_Pressure>
    <Relativistic_Effects>
      <Active>true</Active>
      <Data>
        <Bodies>
          <Item>Earth</Item>
          <Item>Sun</Item>
          <Item>Jupiter Barycenter</Item>
          <Item>Saturn Barycenter</Item>
        </Bodies>
        <Model>General_Relativity</Model>
      </Data>
    </Relativistic_Effects>
    <Yarkovsky>
      <Active>true</Active>
    </Yarkovsky>
    <Poynting_Robertson>
      <Active>false</Active>
    </Poynting_Robertson>
    <Outgassing>
      <Active>false</Active>
      <Data>
        <A1>2.0E-9</A1>
        <A2>1.1E-9</A2>
        <Delta_T>0.0</Delta_T>
        <R0>2.808</R0>
      </Data>
    </Outgassing>
  </Perturbations>
</Propagation>

```

Figure 2 Propagation XML Node

In the Initial Conditions tag, a “.ci” input file has to be specified. This file will provide the initial state and epoch for the propagation. If the Input File tag contains the keyword “Default”, the tool will look for a “.ci” file in the “output” folder with the same ID provided to the executable. This means that the user should modify the content of this tag only if he wants to use a different “.ci” file.

In the Integration tag, all the settings related to the integrator need to be provided. The Integrator tag allows the selection of the integrator. The following integrators have been implemented:

Table 17 Integrators implemented

Integrator	Single/Multi-Step	Step-Size	Integrator Identifier
Runge-Kutta 45	single	variable	Runge_Kutta_45
Dormand Prince 8	single	variable	Dormand_Prince_8
Runge-Kutta 853	single	variable	Runge_Kutta_853
Runge-Kutta 4	single	fixed	Runge_Kutta_4
Runge-Kutta 4 Adapted	single	fixed*	Runge_Kutta_4_Adapted
Gauss-Jackson 8	multi	fixed	Gauss_Jackson_8
Gauss-Jackson 8 Adapted	multi	fixed*	Gauss_Jackson_8_Adapted
Gauss-Jackson 8 Self-Adapted	multi	fixed*	Gauss_Jackson_8_Self_Adapted

*The integration follows a fixed step-size scheme, but for some trajectory arcs (e.g. close to a celestial body) the step-size might be reduced by a factor of 10.

The term “adapted” has been used to distinguish the original integrator scheme from a special algorithm implemented in NEOPROP2 in order to improve the interplanetary trajectory integration. The “adapted” integrators are fixed step-size integrators which are able to reduce the step-size during the integration. For instance, when the NEO exits from the SOI of the Sun, these integrators reduce the step-size by a factor of 10. Then, when the NEO re-enters the SOI of the Sun, the step-size is set back to the initial value. This algorithm has been implemented, because in these arcs the perturbing accelerations become quite big (only few order of magnitude smaller than the main acceleration) and in order to improve the accuracy a smaller step-size is required. On the other hand, along most of the trajectory (when gravity perturbations are not so relevant) larger step-size allow better performances (for more details see [AD3], [AD4] and [AD5]).

The same idea is behind the Gauss-Jackson 8 “self-adapted” integrator. In this case step-size is reduced by a factor of 10 when the ratio between the total perturbing acceleration (due to perturbing gravity fields, solar radiation pressure, etc) and the total acceleration is larger than 0.0001.

The choice of the integration scheme has been left to the user. Depending on the specific problem, the user can select the most suitable integrator. Some guidelines are provided here:

- For the beginning, the recommended integrator is the Gauss-Jackson 8 adapted. It is one of the fastest, even though due to the step-size reduction it is still quite accurate even in case of large local perturbations;
- Variable step-size integrators are recommended in case of multiple close approaches or in case of large perturbations all along the trajectory: in this case the step-size is set depending on the punctual local error;

- Variable step-size are usually slower than fixed step-size integrators, but not always more accurate;
- For most of the NEOs (in particular those without close approaches), the choice of the integrator will affect only the computational time and not the final accuracy;
- For the fixed step-size integrators, the choice of a very big step-size (e.g. larger than 86400 s.) is not recommended.

For the variable step-size integrators the other parameters which have to be specified are the Minimum Step Size, the Maximum Step Size and the Local Tolerance. If the minimum step-size is reached the integration is stopped. This means that the local tolerance specified is too high and cannot be reached with the specified minimum step-size. For fixed step-size integrators only the Maximum Step Size is required, since the minimum one is set equal to the maximum (no local error check is performed).

Moreover an output interval can be selected in order to be outputted in the output file. In case the maximum step size selected would overtake a certain output point, the integrator will come back in the integration recovering the “lost” point. Then the integration will continue from the first position computed before stopping. In this way the output points will not influence the integration process.

The Final Epoch node determines when the propagation will be stopped. The Time Standard parameter can have one of the following values: UTC, TDB, TT, TAI, and GPS (see [RD16]). The Calendar Date tag can be replaced by the Julian Date tag in case the user wants to specify the final epoch as Julian date. Then the XML node becomes:

```
<Final_Epoch>
  <Julian_Date>2600000.5</Julian_Date>
  <Time_Standard>UTC</Time_Standard>
</Final_Epoch>
```

Figure 3 Final Epoch (Julian Date) XML Node

In the Perturbations node only the Third Body Perturbations, the Solar Radiation Pressure and Outgassing are read by this module. The perturbation due to relativistic effects, Poynting-Robertson and Yarkovsky effect are available only for the “complete analysis”. The outgassing perturbation requires the following additional inputs: A1 [AU/day²] (non-gravitational radial acceleration parameter), A2 [AU/day²] (non-gravitational transverse acceleration parameter), A3 [AU/day²] (non-gravitational normal acceleration parameter), Delta_T [day] and R0 [AU], which are described in [AD6]. This perturbation should be activated ONLY for comets.

Another difference regards the list of third bodies considered. In the “fast analysis” all planets and the Moon are considered, while in the “complete analysis” also the four main asteroids (Pallas, Vesta, Hygiea and Ceres) are included.

In the Third Body Perturbations node, the flag “Print_Third_Body_Trajectories” can be specified. If the value is set to “true” then the third body relative positions file is printed (see 6.3.3.2). If this flag is not present in the XML file, by default the flag is set to false (file not printed). The Minimum_Close_Approach_Distance_To_Record allows to modify the minimum distance which defines a close approach (0.005 AU by default, see 6.3.3.3).

```

<Third_Body_Perturbations>
  <Active>true</Active>
  <Print_Third_Body_Trajectories>false</Print_Third_Body_Trajectories>
  <Minimum_Close_Approach_Distance_To_Record>0.1</Minimum_Close_Approach_Distance_To_Record>
</Third_Body_Perturbations>
  
```

Figure 4 Third body trajectory flag node

11.2.2.2.2 Asteroid

The asteroid node is shown here:

```

<Asteroid>
  <Mass>2.782618569E10</Mass>
  <Absolute_Magnitude>19.1</Absolute_Magnitude>
  <Diameter>270.0</Diameter>
  <Bulk_Density>2700.0</Bulk_Density>
  <Surface_Density>2000.0</Surface_Density>
  <Thermal_Conductivity>0.01</Thermal_Conductivity>
  <Thermal_Capacity>680.0</Thermal_Capacity>
  <Albedo>0.33</Albedo>
  <Infrared_Emissivity>0.9</Infrared_Emissivity>
  <Spin_Axis_Obliquity>0.0</Spin_Axis_Obliquity>
  <Rotational_Period>30.4</Rotational_Period>
</Asteroid>
  
```

Figure 5 Asteroid XML Node

All the physical properties of the NEO are collected in the next table, along with the default values, units and the perturbations affected by them:

Table 18 Asteroid Physical Properties

Parameter	Perturbation for which is required	Default Value
Mass	All	Bulk density * Pi / (6.0 * diameter ³) [kg]
Absolute_Magnitude	All	- [-]
Diameter	All	$1329000.0/\sqrt{(0.1)*10^{(-0.2*Magnitude)*albedo^{-0.5}}}$ [m] [RD17]
Bulk_Density	Yarkovsky	2700.0 [kg/m3] [RD19]
Surface_Density	Yarkovsky	2000.0 [kg/m3] [RD19]
Thermal_Conductivity	Yarkovsky	0.01 [W/m/K] [RD19]
Thermal_Capacity	Yarkovsky	680.0 [J/K/kg] [RD19]
Albedo	Yarkovsky, SRP, Poynting-Robertson	0.33 [-] [RD19]
Infrared_Emissivity	Yarkovsky	0.9 [-] [RD19]

Spin_Axis_Obliquity	Yarkovsky	0.0 [°] [RD19]
Rotational_Period	Yarkovsky	10.0 [h]

At least the Diameter or the Absolute Magnitude must be provided by the user. If this is not the case, the tool will not start the trajectory propagation. For all the other parameters, the user can write in the XML file "Default", in this way the tool will use the default values provided in Table 18. For the "fast analysis" only the parameters involving the SRP must be specified.

All the tags should be kept in the XML file (in case all the parameters can be set to "Default"), in order to avoid problems while running the "complete analysis".

11.2.2.3 Step-by-Step Procedure

In case the user needs to create from scratch a new test case, a valid test case folder structure should be generated as presented and explained in section 9.2. The 2013 TP4 NEO will be used for this example. The ID mentioned in chapter 6.3, which is used to generate input and output files with the proper naming convention, can be set to 2013_TP4, even though any other ID would work too. All computations do not take into consideration the selected ID, but just the data stored in the files.

The following steps are required to generate a valid test case. If a valid test case is already available, because previously created as described in section 11.2.1.3, the user should jump to step 3 and ignore previous two steps.

1. Create a new folder, which will contain all the input/output files. For this example the folder can be named 'Test'.
2. Create inside the 'Test' folder two subfolders: 'input' and 'output'.
3. Place inside the 'input' folder the necessary input file:
 - a. **2013_TP4.xml**, which contains the input parameters for the numerical module. The 'template.xml' file, present in the 'doc' folder, can be used as starting point for the creation of the aforementioned file. A copy of this file should be created and placed into the 'input' folder and then renamed into 2013_TP4.xml. Then it should be edited in order to use the correct settings for the current object, especially physical parameters as mass (or the gravity parameter mu) and the radius (or the magnitude). For this test, the following modifications must be applied:
 - '<Mass>Default</Mass>' should be replaced by '<Mass>2192127.608921</Mass>'
 - '<Absolute_Magnitude>25.0</Absolute_Magnitude>' should be replaced by '<Absolute_Magnitude>27.304</Absolute_Magnitude>'

The meaning of all the other parameters can be found in section 11.2.2.2.1.

4. Place inside the 'output' folder the necessary input file:
 - a. **2013_TP4.ci**, which contains the initial state of the NEO. If the user wants to load the initial state from another input file (with a different name or placed in a different folder), the XML node '<Input_File>' should contain the absolute path of the '.ci' file to be used (as already mentioned in section 11.2.2.2.1).

Alternatively to the procedure just described, the user could make use of the GUI to generate a new test case, including all the requested input files, as explained in section 11.2.4.2). Furthermore, The GUI could be used to execute the test case, as showed in paragraph 11.2.4.4.

If the user wants to run the test case from command-line, the “NEOPROP_prop_fast.exe” executable should be invoked with the proper set of arguments as described in Table 16. To open the command prompt window, run cmd.exe on Windows or use the keystroke Ctrl+Alt+t to open the terminal for Linux. The command-line call is showed below:

```
"NEOPROP_installation_folder_absolute_path\bin\NEOPROP_prop_fast.exe"  
"Test_folder_absolute_path\Test" "2013_TP4"
```

A similar call is required to run the delivered ‘Test_Case_0030’, present in the ‘Test_cases’ folder. The following call must be used:

```
"NEOPROP_installation_folder_absolute_path\bin\NEOPROP_prop_fast.exe"  
"NEOPROP_installation_folder_absolute_path\Test_cases\Test_Case_0030" "99942"
```

It is worth noticing that the first two arguments are the absolute paths of the executable and of the ‘Test’ folder. Moreover, the same list of inputs and arguments could be used to run the “NEOPROP_prop_complete.exe”. In this way additional perturbations can be used for the trajectory propagation.

At the end of the execution the following files (see 6.3.3.1 and 6.3.3.3 for the files formats) are generated in the ‘output’ folder:

- 2013_TP4_nominal_trajectory_fast.txt
- 2013_TP4_close_approaches_fast.txt

In case an error occurs during the execution of this module, a file containing additional information about the error is created: 2013_TP4_errorLog_prop_fast.txt (as mentioned in paragraph 6.3.3.5).

11.2.2.4 Output

The outputs of this task are text files:

- “_nominal_trajectory_fast.txt” file with the nominal trajectory as function of the time (see 6.3.3.1);
- “_close_approaches_fast.txt” file with the all the close encounters to any planet and the MOID (see 6.3.3.3).

11.2.3 Numerical Module (complete analysis)

11.2.3.1 Introduction

This task is intended to be used for a complete risk assessment of a NEO. Starting from the orbit determined by the Analytical module and from a set of VAs, the nominal trajectory will be propagated and close approaches and virtual Earth impactors computed. All the provided VAs will be also propagated in order to give to the user some more information about the sensitivity of the MOID due to uncertainties in the initial

conditions. If requested by the user, the Numerical module can compute with a Monte Carlo run a user defined number of VAs initial states using the provided uncertainties of the nominal state as 1- σ variation. In this way a mix of random VAs and VAs along the LOV can be propagated, as already discussed and proposed in [AD5].

11.2.3.2 Input

The inputs required to run this module are of two different types:

- the list of arguments to invoke the executable “NEOPROP_prop_complete.exe” (see 9.4);
- an XML file with the settings required to initialize the propagator (see 6.3.1.3 and 9.2 for the file name convention), which defines the ‘.ci’ and ‘.va’ files to be used.

As already explained in 11.2.2.2, the XML file is shared with the “fast analysis”. For this reason, only those parts not discussed yet will be assessed now.

11.2.3.2.1 Propagation

The structure of the “Propagation” XML node has been already discussed in 11.2.2.2.1. The main differences regard the different set of perturbations available. For this analysis, relativistic effects, Yarkovsky and Poynting-Robertson perturbations can be used. While for the last two it is enough to set their flag to “true”, the relativistic effects perturbation requires some more input.

The user shall provide the list of bodies considered for the computation of the relativistic effects (as shown in Figure 2). Then two different models can be specified: the “General_Relativity” or the “PPN” models. In the first case, the user does not need to provide any additional input. For the PPN model the user shall provide the list of PPN parameters [RD18], as shown in the following figure:

```

<Relativistic_Effects>
  <Active>true</Active>
  <Data>
    <Bodies>
      <Item>Earth</Item>
      <Item>Sun</Item>
      <Item>Jupiter Barycenter</Item>
      <Item>Saturn Barycenter</Item>
    </Bodies>
    <Model>PPN</Model>
    <Rel_Beta>1.0</Rel_Beta>
    <Rel_Gamma>1.0</Rel_Gamma>
    <Rel_Alpha1>1.0</Rel_Alpha1>
    <Rel_Alpha2>1.0</Rel_Alpha2>
    <Rel_Alpha3>1.0</Rel_Alpha3>
    <Rel_Zeta1>1.0</Rel_Zeta1>
    <Rel_Zeta2>1.0</Rel_Zeta2>
    <Rel_Zeta3>1.0</Rel_Zeta3>
    <Rel_Zeta4>1.0</Rel_Zeta4>
    <Rel_W>1.0</Rel_W>
  </Data>
</Relativistic_Effects>
  
```

Figure 6 PPN Relativistic Effects Model Node

Moreover, this propagator includes the gravitational effects of the following massive asteroids: Vesta, Ceres, Hygea and Pallas.

11.2.3.2.2 Asteroid

The structure of the “Asteroid” node has been already widely discussed in 11.2.2.2.2. For the “complete analysis” all the parameters listed in Table 18 might be used, depending on the selected perturbations.

11.2.3.2.3 Virtual Asteroids

The Virtual Asteroid tag is optional and is required only if the user wants to propagate virtual asteroids trajectories and assess the impact risk of the NEO. In this case, a valid input file with a set of VA initial states and/or the inputs for the creation of random asteroids must be provided. The file format of the input file must be the same as described in 6.3.2.2. The creation of such file should be left to the Analytical module. If the Input File tag contains the keyword ‘Default’, the tool will look for a “.va” file in the “output” folder with the same ID provided to the executable. If the Input File tag is not present, the tool will not try to load VAs from file.

Moreover, the number of random VAs to be created and propagated by the Numerical module can be specified. If the Random VA tag is missing, no random VAs will be computed. The creation of random VAs is totally independent from the VAs imported from file. The Seed field allows the user to reproduce the VAs and indeed the same results, since it is the integer value used to initialize a pseudorandom number generator.

```
<Virtual_Asteroids>
  <Input_File>Default</Input_File>
  <Random_VA>
    <Number>5000</Number>
    <Seed>1</Seed>
  </Random_VA>
</Virtual_Asteroids>
```

Figure 7 Virtual Asteroids XML Node

Since the propagation of VAs is parallelized, the user can decide how many thread to assign to this task in Number_Of_Threads tag. If a number bigger than the number of available threads is specified, all the available ones will be used.

```
<Virtual_Asteroids>
  <Input_File>Default</Input_File>
</Virtual_Asteroids>
<Number_Of_Threads>6</Number_Of_Threads>
```

Figure 8 Number of threads XML Node

If a valid ‘.va’ file is found and/or random VAs are defined, their trajectories are propagated. If a virtual Earth impactor is found (distance to Earth below 1.5 Earth’s radius), the impact probability is evaluated as explained in [RD22] and [AD6] with the following formula:

$$P_i = \int_0^{r_{\oplus}} PDF(r) dr = CDF(r_{\oplus}) - CDF(0) = \frac{1}{2} \left[\operatorname{erf} \left(\frac{r_{\oplus} - \bar{d}}{\sigma\sqrt{2}} \right) - \operatorname{erf} \left(\frac{-\bar{d}}{\sigma\sqrt{2}} \right) \right] \quad \text{Eq. 1}$$

where CDF denotes the cumulative density function, \bar{d} is the mean distance from the Earth, σ is the standard deviation of the perigee distances from the Earth, and r_{\oplus} is the radius of the Earth.

The impact probability is then used to compute Torino and Palermo scale values.

11.2.3.3 Step-by-Step Procedure

In case the user needs to create from scratch a new test case, a valid test case folder structure should be generated as presented and explained in section 11.2.2.3. The 2013 TP4 NEO will be used for this example too and the ID mentioned in chapter 6.3 is set to 2013_TP4.

The differences between the setup needed to run the “NEOPROP_prop_fast.exe” and the “NEOPROP_prop_complete.exe” are explained here:

- The ‘<Virtual_Asteroids>’ tag now must be part of the input XML file. In the fast analysis, even if present, this tag is always ignored.
- **2013_TP4.va**, which contains the virtual asteroids initial states, must be placed in the ‘output’ folder. If the user wants to use a different file (with a different name or placed in a different folder), the XML node ‘<Input_File>’ (within the ‘<Virtual_Asteroids>’ tag) should contain the absolute path of the ‘.va’ file to be used (as already mentioned in section 11.2.3.2.3)

At this point, it is possible to run the test case invoking the “NEOPROP_prop_complete.exe” executable with the proper set of arguments as described in Table 16. To open the command prompt window, run cmd.exe on Windows or use the keystroke Ctrl+Alt+t to open the terminal for Linux. The command-line call is showed below:

```
"NEOPROP_installation_folder_absolute_path\bin\NEOPROP_prop_complete.exe"
"Test_folder_absolute_path\Test" "2013_TP4"
```

A similar call is required to run the delivered ‘Test_Case_0040’, present in the ‘Test_cases’ folder. The following call must be used:

```
"NEOPROP_installation_folder_absolute_path\bin\NEOPROP_prop_complete.exe"
"NEOPROP_installation_folder_absolute_path\Test_cases\Test_Case_0040" "99942"
```

It is worth noticing that the first two arguments are the absolute paths of the executable and of the ‘Test’ folder.

At the end of the execution the following files (see 6.3.3.1, 6.3.3.3 and 6.3.3.4 for the files formats) are generated in the ‘output’ folder:

- 2013_TP4_nominal_trajectory_complete.txt
- 2013_TP4_close_approaches_complete.txt
- 2013_TP4_VAs_final.txt

In case an error occurs during the execution of this module, a file containing additional information about the error is created: 2013_TP4_errorLog_prop_complete.txt (as mentioned in paragraph 6.3.3.5).

11.2.3.4 Output

The outputs of this task are text files:

- “_nominal_trajectory_complete.txt” file with the nominal trajectory as function of the time (see 6.3.3.1);
- “_close_approaches_complete.txt” file with the all the ‘nominal’ close encounters to any planet, virtual Earth impactors and risk assessment (see 6.3.3.3) consisting in asteroid mass, kinetic energy, impact probability, Palermo and Torino scale values;
- “_VAs_final.txt” file with the some final statistics of the VAs propagated trajectories (see 6.3.3.4).

11.2.4 GUI

11.2.4.1 Introduction

As already mentioned, the GUI has been developed for testing purposes. Nevertheless it is delivered together with the other executables. It does not add any additional feature, but it simplifies the life of users. In this chapter the three main functionalities are presented: creating a new scenario, running it and evaluating the results.

Before introducing them, some definitions are provided in order to clearly define the different sections of the GUI.

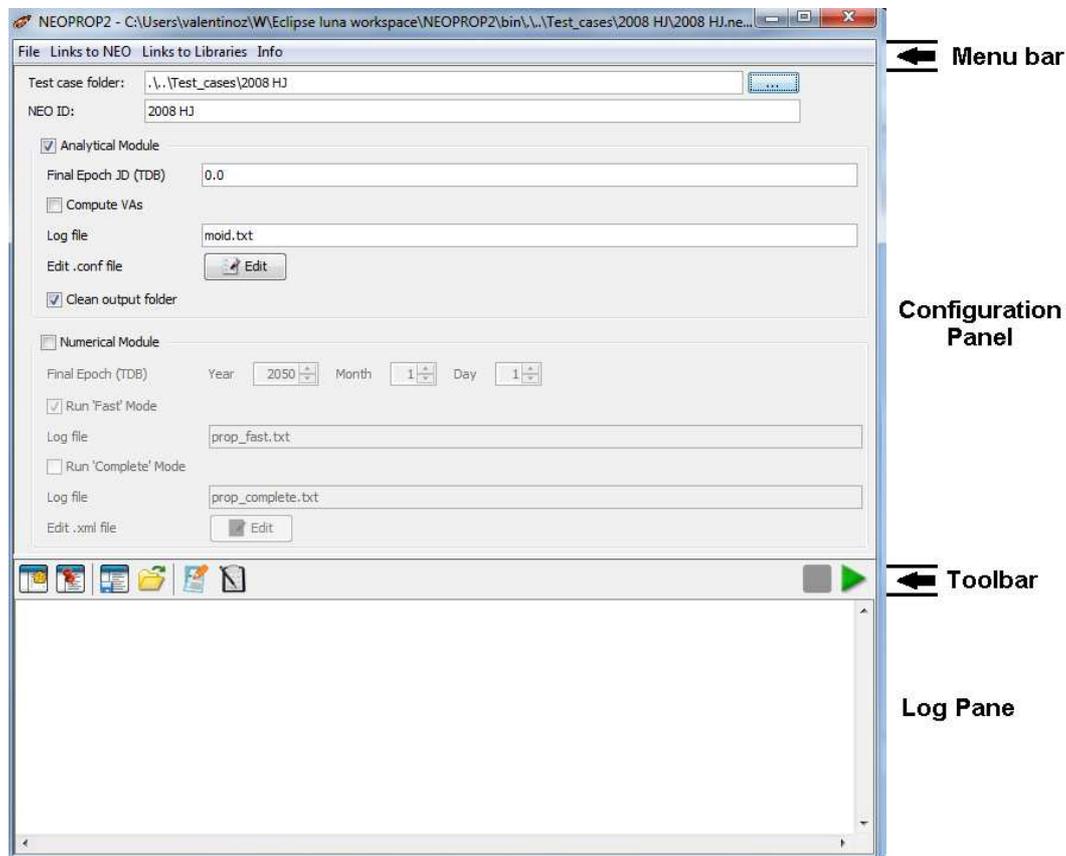


Figure 9 GUI's screenshot and structure

The menu bar contains the following submenus:

- **File**, which allows the user to create a new scenario (New), load an existing one (Open), reloading (Reload) or saving (Save) the current scenario. Moreover, with the 'Open Scenario folder' entry it is possible to open an instance of explorer directly inside the Scenario folder, while with the 'Backup' entry a copy of all the output files can be created; a new folder with the provided name is created in the 'output' folder. At the end of the menu, the 4 latest loaded scenarios are showed in order to easily reaccess them.
- **Links to NEO** provides some useful internet links to the MPC, JPL, ESA SSA, NEODyS and Sentry pages. In the last section links for the current NEO to the NEODyS and Sentry pages are showed. If the NEO is inserted in the risk list, links to the NEODyS and Sentry risk pages are showed too, instead of being greyed out.
- The **Links to Libraries** menu provides some useful internet links to the internet pages where most of the NEOPROP2 libraries (already defined in 6.2) are taken/updated from. In this way, the user can easily check these pages to find out if an update of a certain library is necessary.
- The **Info** menu contains the link to this document and some general information about the tool itself.

The configuration panel allows the user to modify settings of the current scenario, which are introduced and explained in 6.3.1.4. The current values, displayed in this panel, are

used once the 'Run' button is pressed, even if the scenario has not been saved after modifying the settings. Of course, if the GUI is closed without saving, all modified values are lost. This is because the GUI configuration file is used only to initialize the GUI and not as input to the other executables. For the same reason, changing the 'NEO ID' does not automatically trigger a renaming of all the input/output files, but it affects only the list of arguments (see 9.4) passed to the executables. In this panel the user can select also which executables have to be run (as explained in 11.2.4.3).

Two 'Edit' buttons are available, one in the Analytical Module section and the other one in the Numerical Module section. These buttons allow to open the default text editor in order to edit respectively the '.conf' and the '.xml' files.

The toolbar allows the user to run (and stop) any NEOPROP2 executable, to create a scenario summary or results comparison and also to save or load the content of the log pane. The following buttons are part of the toolbar:

-  can clear the log pane text;
-  can hold the log pane at a certain fixed position, while the output stream is continuously flushed into the screen;
-  allows to load into the log pane a text (e.g. log file) or html file (e.g. report);
-  allows to save the content of the log pane into a text or html file;
-  creates a scenario summary as described in 11.2.4.5;
-  allows to compare the scenario results with the values computed by NEODyS and Sentry (see 11.2.4.5 for more details);
-  is available only when an executable is running and it allows to stop its execution;
-  is available only when no executable is running and it allows to run the scenario.

The log pane allows the user to visualize in real time (in text format) the output stream of a running executable. In this way the user is informed about the current status of a certain action/analysis. This pane is also used to visualize (in html format) the scenario report and results comparison, as explained in 11.2.4.5.

11.2.4.2 Scenario Creation

A new Scenario can be created from File->New. The 'New Scenario' dialog (showed below) allows the user to collect with few clicks all the inputs required to set up a new test case. All the requested input files (.conf, .xml and observations) are automatically created. Of course, not all the settings stored in the .conf and .xml can be modified from the GUI, but most of the applications should be able to run successfully with the default values.

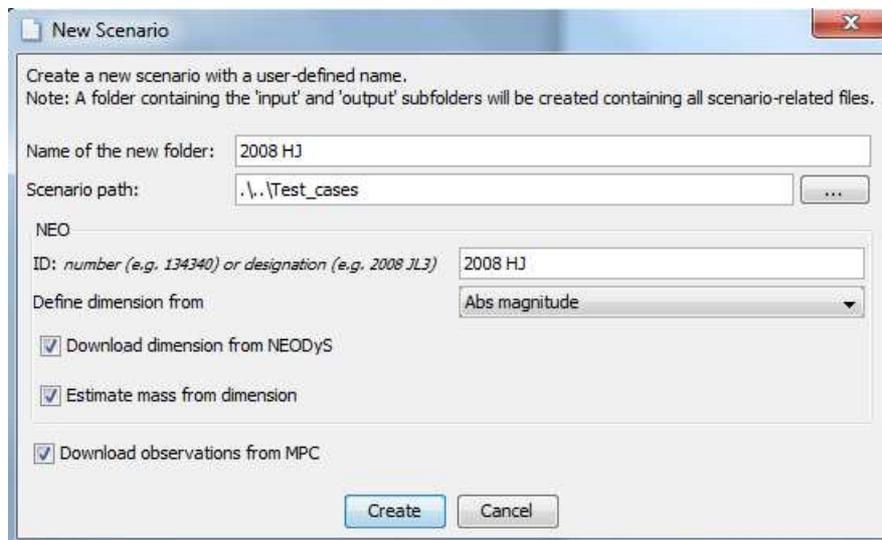


Figure 10 Screenshot of the GUI dialog to create a new scenario

Once the 'Create' button in the 'New Scenario' dialog is pressed, a new scenario is created: a new folder with the provided name is generated and placed at the specified location. The 'input' and 'output' subfolders are created too, together with the '.neoprop' GUI configuration file.

The NEO ID has to be provided in order to determine filenames and to access online resources. The ID should be the same as used by the MPC website [RD11], including eventual empty spaces.

The dimension of the NEO has to be provided by specifying the absolute magnitude [-] or the radius [m] of the object. If it is not known, it can be downloaded from the NEODyS website at page '<http://newton.dm.unipi.it/~neodys2/epoch/ID.eq0>'. Similarly the mass [kg] of the NEO can be directly provided by the user or estimated from its dimension, assuming an albedo of 0.154 (in case the absolute magnitude is used) or assuming a density of 2600 kg/m³ (in case the radius is used), by means of formulas showed in Table 18.

Once the creation process is completed, a copy of the 'template.conf' and 'template.xml' files is placed in the newly created 'input' folder. These files are renamed accordingly to the specified ID. Moreover, the default dimension and mass of the NEO present in the template files are replaced by the user defined values. In case the provided ID is recognized as belonging to a comet (if contains '/'), then the 'is_neo_check' and 'constrain_solution' flags are set to false. In this way the user does not need to manually edit these files.

If the 'Download observations from MPC' checkbox is selected, the tool tries to download the observation file from the MPC website, using the provided ID. If radar and/or optical observations are found, they are stored in dedicated files ('ID.obs' and 'ID.rad') with the format defined in 6.3.1.2.

In case the NEO's ID provided by the user is not known by MPC or by NEODyS or if a connection problem occurs, an error message is raised.

11.2.4.3 Scenario Import

In case a scenario was previously created, it is possible to import it from File->Open and run it again. Once the 'Open' button is pressed the user needs to select a '.neoprop' file previously created (usually located in the test case top folder). Then the GUI can load all the settings showed in the configuration panel and stored in the '.neoprop' file.

After loading all the settings, it is important that the user checks that the displayed test case folder is still correct. When moving a test case folder, the absolute path stored in the GUI configuration file is not updated, for this reason when reloading that test case, the folder path should be modified. If the user needs to modify the test case folder and the NEO ID (or any other GUI setting), it is recommended to save the scenario in order to overwrite the stored settings.

11.2.4.4 Scenario Execution

Once a new scenario has been created (or an old one has been loaded from File->Open), it is possible to run NEOPROP2. The 'Run' button runs in series all the tasks that have been selected by the user in the configuration panel:

- Analytical module;
- Numerical module 'Fast' mode;
- Numerical module 'Complete' mode.

The final epoch specified in the configuration panel for the analytical and numerical module is always used by the executables directly (in case of the analytical module it is passed as argument to the executable) or indirectly (it is written in the .xml file read by the numerical module).

The output stream for each task is saved into a log file (whose name is user defined and showed in the 'Log file' field), placed in the output folder.

11.2.4.5 Scenario Report

Users can generate a scenario summary and/or compare the results, obtained by NEOPROP2, with the values reported by NEODyS and Sentry. In both cases results are separated in three sections, one per task, and for each of them the last time the task has been successfully executed is reported. In the results of the numerical module, also the minimum close approach distance to be recorded (see end of paragraph 11.2.2.2.1) is written. This value determines which close approaches are recorded by NEOPROP2 and which ones are read from NEODyS and Sentry for the results comparison. If this value is too small, only very close approaches are noted.

The scenario summary functionality collects all the information from the output files stored in the output folder and writes them into the log pane. In this way the user gets a rapid overview of the results without the need of accessing manually all the files. The results sorted per task which are reported by this functionality are listed in the next table.

Table 19 Results collected by the scenario summary functionality

Analytical Module
Equinoctial elements

Keplerian elements
MOID
Numerical 'Fast' Module
Nominal close approaches
Numerical 'Complete' Module
Nominal close approaches
Earth virtual impactors (if found) and relative risk impact assessment

The scenario results comparison functionality allows the user to compare the results generated by NEOPROP2 with those reported by the NEODyS and Sentry systems. The internet pages, where the required information can be found, are downloaded by the GUI and saved (as html or txt files) in the 'reference' folder, which is created in the output folder. These files are then parsed in order to collect the data required for the results comparison. The following webpages are used:

- <http://newton.dm.unipi.it/~neodys2/epoch/ID.eq0>, NEODyS near middle of observational arc orbital details;
- <http://newton.dm.unipi.it/~neodys2/epoch/ID.eq1>, NEODyS near present day orbital details;
- <http://newton.dm.unipi.it/neodys/index.php?pc=1.1.0&n=ID>, NEODyS NEO's summary;
- <http://newton.dm.unipi.it/neodys/index.php?pc=1.1.8&n=ID>, NEODyS NEO's close approaches;
- <http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=ID;old=0;orb=0;cov=0;log=0;cad=1#cad>, Sentry NEO's summary (including close-approaches);
- <http://neo.jpl.nasa.gov/risks/ID.html>, Sentry NEO's risk page.

Both first two files contain the NEO's orbital elements. Only those computed at the epoch closest to the one used by NEOPROP2 are taken as reference for the NEO's orbital elements

It is worth emphasizing that in case the format of any of the above listed pages changes, the GUI may not be able to collect all the information used during the results comparison.

For each task, the log pane shows the values obtained by NEOPROP2 and the 'reference' values computed by NEODyS and Sentry. Then the NEOPROP2 values are compared to the reference ones. It is worth noticing that NEODyS does not include comets in its database. Therefore, in case of a comet, results can be compared only to Sentry and the 'Download dimension from NEODyS' feature (showed in Figure 10) cannot be used.

11.2.5 Demo

In this chapter a short demonstration on how to create and run a new scenario with the GUI is given.

First, the user needs to start the GUI, using the 'Neoprop2.jar' file in the 'bin' folder (in case the user has an installed Java environment) or the 'Neoprop2_local_jre.bat', which uses the delivered Java environment present in the 'jre' folder. Then, the user should select the 'File->New' menu item. The 'New scenario' dialog appears as showed in Figure 10. The user needs to insert the name of the new folder which will host the scenario (e.g. '2008 HJ') and the path where it will be saved. Afterwards, information about the NEO has to be provided:

- ID: e.g. '2008 HJ' (the empty space is important!)
- Define dimension from: 'Abs magnitude', this usually is more precise, especially if it is downloaded from the NEODyS website
- Download dimension from NEODyS: checked, in this way the absolute magnitude will be taken directly from NEODyS. If not selected, the user can specify its own value
- Estimate mass from dimension: checked, in this way the mass is estimated directly from the absolute magnitude (or radius) assuming an albedo of 0.154.

The 'Download observations from MPC' should be selected and allows the user to retrieve all the observations from the MPC database. Then the 'Create' button can be pressed.

If no error is reported (as could happen in case an internet connection is not available or an unknown NEO ID is inserted, etc), a valid test case folder ready to be used is created (as explained in 11.2.4.2).

At this point, the user can run the analytical module to get the orbital elements of the NEO and afterwards the Numerical Module (fast analysis) can be run to evaluate close approaches. If the user is interested in assessing the impact risk, the 'Compute VAs' checkbox has to be selected, before running the analytical module, in order to create a set of virtual asteroids. Then the complete numerical analysis will be able to propagate them and evaluate eventual virtual impactors.

If the user wants to overwrite some of the default values (e.g. to customize the set of perturbations to be considered, to improve the physical properties of the NEO, etc) the .conf and .xml files have to be manually modified.

Appendix A – Reference Frames

Here a brief overview on the most used (in the GUI as well as in the source code) and important reference frames is given.

■ Inertial J2000 (Inertial_J2000)

This very famous and common inertial frame is defined with the Earth's Mean Equator and Equinox at 12:00 terrestrial time on the 1st of January 2000. The x-axis is aligned with the mean equinox. The z-axis is aligned with the Earth's spin axis or celestial north pole. The y-axis is rotated by 90° East about the celestial equator.

This is considered as the “real” inertial reference frame.

■ Mean of Date (Inertial_MOD)

If starting from Inertial_J2000 we don't consider the effect of the precession of the ecliptic (slow secular in the ecliptic's orientation) we get the so called Inertial Mean of Date reference frame.

■ True of Date (Inertial_TOD)

If starting from Inertial_MOD we don't consider the effect of the ecliptic nutation (small oscillations in the Earth's rotation axis) we get the so called True of Date reference frame.

■ Planetocentric (PCPF)

If starting from Inertial_TOD we don't consider the variability of the sidereal time due to the accumulated effects of precession and nutation in the right ascension then we get the Planetocentric reference frame.

■ Body Frame

The body frame has the origin in the center of gravity of the spacecraft. The X_b axis lies in the symmetry plane of the S/C and points forward. The Z_b axis also lies in the symmetry plane, but points downwards. (It is perpendicular to the X_b axis.) The Y_b axis can be determined using the right-hand rule.

Appendix B – List of implemented algorithms

Here a list of the implemented algorithms grouped by task and module.

Task	Algorithm	NEOPROP2 module
Preliminary Orbit Determination	Gauss method	Analytical
Orbital improvement	LLS method	Analytical
Analytical MOID computation	Sitarski method	Analytical
VAs computation	LOV + MC	Analytical + Numerical
Trajectory Integrator	GJ8, RK4, RK853, RK45, DP8	Numerical
Trajectory perturbations	SRP, 3 rd -bodies, outgassing, relativistic effects, Earth and Sun J2, Yarkovsky and Poynting-Robertson	Earth and Sun J2, Yarkovsky and Poynting-Robertson only in Numerical
Impact probability	Non-Linear analysis (probability density function is assumed to be Gaussian)	Numerical