WebGeocalc and Cosmographia: Modern Tools to Access OPS SPICE Data

Boris V. Semenov[1]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109-8099, USA

This paper provides a brief introduction to two modern ways of accessing the capabilities of the SPICE space mission geometry system – the WebGeocalc tool using a client-server architecture and providing a simple GUI and a Web-based API interfaces, and the SPICE-enhanced Cosmographia program used to visualize space missions based on SPICE data.

I. Nomenclature/Acronyms

API	=	Application Programming Interface
CK	=	C-matrix Kernel
CMOD	=	Celestia 3D File Format
CSV	=	Comma Separated Values
DEC	=	Declination
DSK	=	Digital Shape Kernel
DSN	=	Deep Space Network
DSS	=	Deep Space Station
ESA	=	European Space Agency
ESAC	=	European Space Astronomy Centre
FK	=	Frames Kernel
FOV	=	Field-Of-View
GUI	=	Graphical User Interface
IK	=	Instrument Kernel
JPL	=	Jet Propulsion Laboratory
JSON	=	JavaScript Object Notation
LSK	=	Leapseconds Kernel
NAIF	=	Navigation and Ancillary Information Facility
NASA	=	National Aeronautics and Space Administration
OPS	=	Operations
PCK	=	Planetary Constants Kernel
PDS	=	Planetary Data System
PNG	=	Portable Network Graphics
RA	=	Right Ascension
REST	=	REpresentational State Transfer
SCLK	=	Spacecraft Clock Kernel, also Spacecraft Clock
SPICE	=	Spacecraft Planet Instrument C-matrix Events
SPK	=	Spacecraft Planet Kernel
TDB	=	Barycentric Dynamical Time
TDT	=	Terrestrial Dynamical Time
TGO	=	2016 ExoMars Trace Gas Orbiter
UTC	=	Universal Time Coordinated
WGC	=	WebGeocalc

[1] Senior Software Engineer, Navigation and Ancillary Information Facility (NAIF) Group, Jet Propulsion Laboratory, 4800 Oak Grove Dr., Pasadena, CA 91109-8099 / Mail Stop 301-121

II. Introduction

For more than two decades navigation and other ancillary data from most US and international planetary science missions have been packaged using "SPICE" (Spacecraft, Planet, Instrument, Camera-matrix, Events) system data files (a.k.a. SPICE kernels) and, in conjunction with SPICE Toolkit software used by scientists and engineers to compute observation geometry in various ground system tools ranging from mission planning and analysis applications to data production pipelines to science analysis tools. The traditional way for accessing SPICE data is by downloading necessary SPICE kernels to a user's workstation, installing the SPICE Toolkit software available from NAIF, and writing an application calling APIs from the SPICE Toolkit library to compute numeric geometric parameters of interest. While this approach did and still does provide the greatest flexibility in implementing geometric computations of interest, it proved to be complicated for users with little programming abilities, required data to be always copied to the users' workstations, and lacked any out-of-the-box visualization capabilities. To address these shortcomings NAIF developed the WebGeocalc (WGC) tool and extended the publicly available Cosmographia program to use SPICE. Employing these two new tools in mission operations enables easier access to SPICE computations and SPICE-based visualizations for a wider variety of mission personnel.

III. Traditional SPICE Use

A. SPICE Overview

SPICE [1] is a system of navigation and ancillary data developed the Navigation and Ancillary Information Facility (NAIF) group at the Jet Propulsion Laboratory (JPL). The purpose of SPICE is to assist scientists and engineers involved in modeling, planning and executing activities needed to conduct space exploration missions. The two primary parts of SPICE are data files, called "kernels", and software, called the SPICE Toolkit.

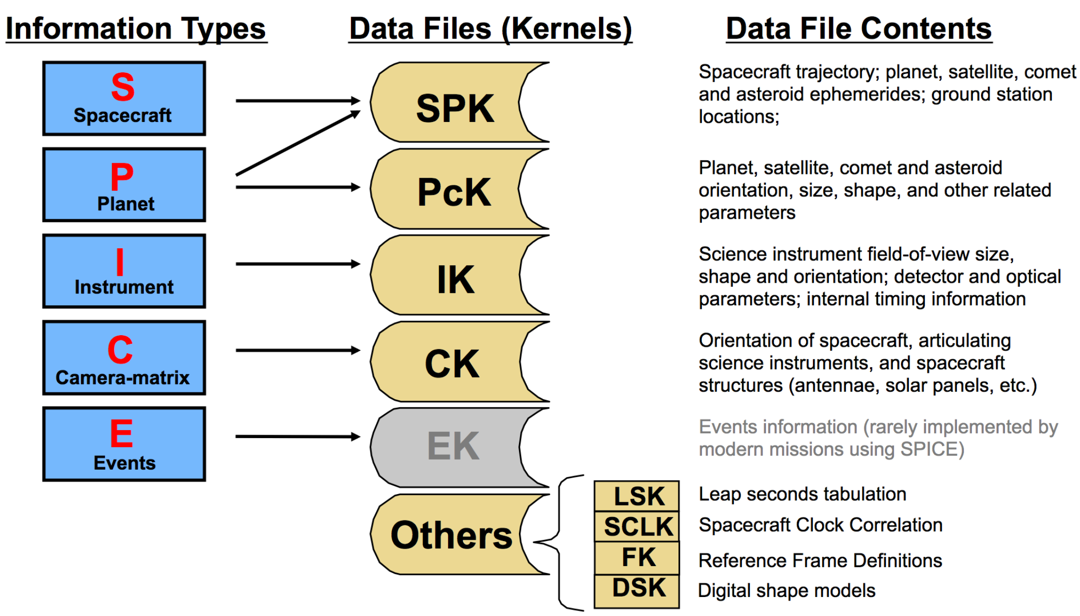

Fig. 1 SPICE Data Types.

Different information types in SPICE are stored in different kernels (Fig. 1). Kernels come in two basic forms., binary and text, are structured and formatted for easy access and correct use, and are portable between various computer environments. The SPICE Toolkit, available in multiple languages (FORTRAN, C, IDL, MATLAB, Java/JNI) for most common computer environments (Linux, OS X, Windows, Solaris), is the mechanism to use SPICE data. The core of the SPICE toolkit is a library providing API interfaces for accessing data from kernels and for computing various derived space geometry quantities, such as positions and frame transformations, using this data. The toolkit also includes utility programs for ancillary tasks such as creating, summarizing, and altering kernels, and extensive documentation covering all aspects of its functionality. Historically the toolkit does not include any GUI applications, neither for computing derived quantities nor for SPICE-based graphics/visualizations.

B. Traditional SPICE Use Scenario

In a traditional SPICE use scenario a user must download all necessary kernels to his/her workstation, install the SPICE toolkit, and write an application that links to the SPICE library and reads the locally installed kernels (Fig. 2).

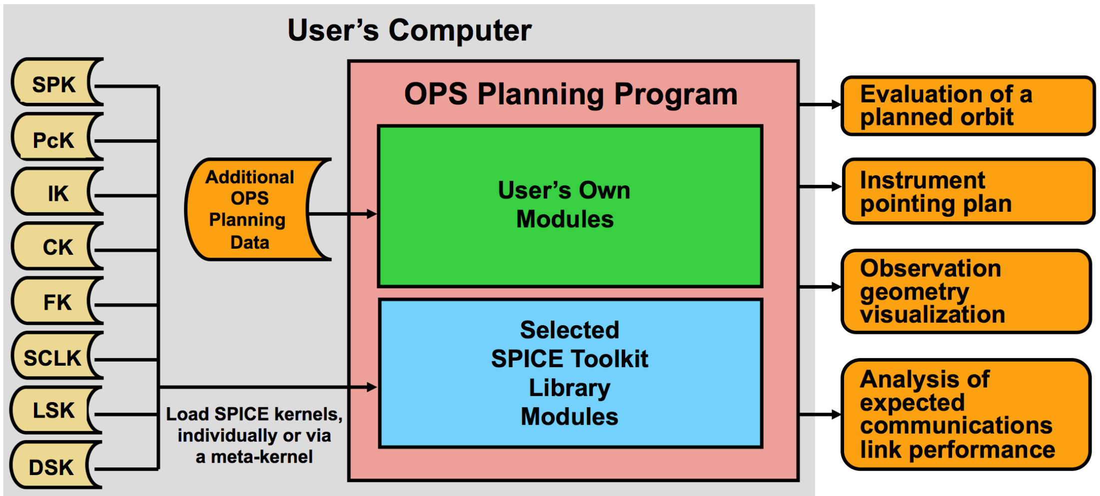

Fig. 2 Traditional SPICE-based Application Architecture.

While this approach provides the greatest flexibility in getting access to geometric computations from SPICE, it is difficult for users with few programming skills who would rather use a simple GUI interface requiring no programming effort. Since even more programming expertize is needed to create a professionally looking visualization application, such a task is likely out of reach for even medium skilled programmers. The fact that all needed kernels must be available locally puts the burden on the users to copy them to their workstations and to carefully manage them, which sometimes is a challenging task.

IV. WebGeocalc

A. Introduction

NAIF created WebGeocalc (WGC) to address the needs of those who can't program and to provide a SPICE interface that does not require kernels to be located on a user's workstation. WGC provides a Graphical User Interface (GUI) [2] and Programmatic Interface (API) to a SPICE server running a geometry computation engine and configured to access collections of operational SPICE kernels (Fig. 3).

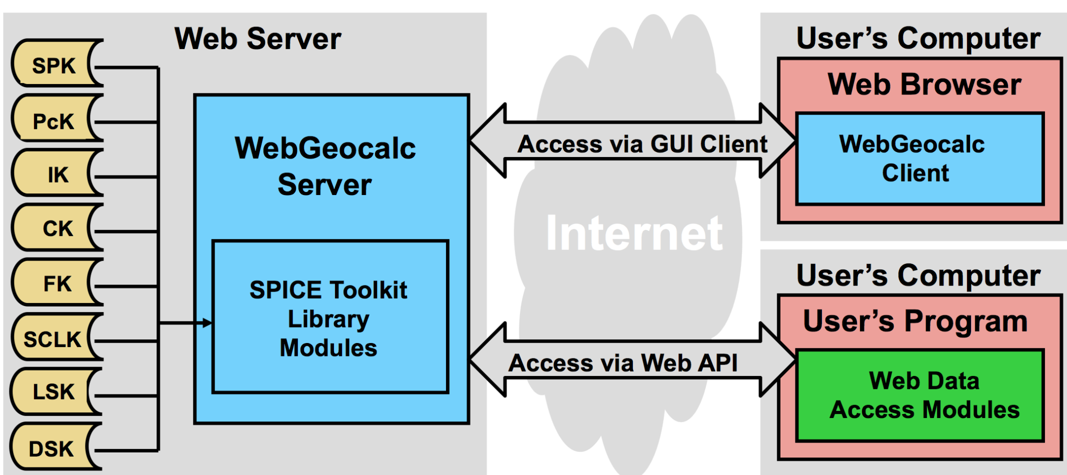

Fig. 3 WGC Architecture.

Via the GUI interface WGC makes it “easy” to do many kinds of SPICE computations, eliminating the need to write a program using the SPICE Toolkit software, and allowing instead just using a web browser to enter inputs into forms and have results, including plots, appear in the browser window. By means of this simple interface WGC can support mission operations many ways, including helping to check computations done in a custom SPICE-based program under development, helping to quickly solve a one-off space geometry problem, and allowing those unable to write SPICE-based programs to nevertheless make some kinds of space geometry computations

Via the API interface (available summer 2018) WGC allows any application that needs to compute SPICE-based geometry parameters from the set supported by WGC to do so using standard Web data access APIs. Through such APIs an application can submit calculation requests to and retrieve calculation results from the WGC server using RESTful structured URLs with JSON request and returns payloads. Using the API interface eliminates the need for applications to be linked to the SPICE toolkit and to have kernels available locally on the workstations where they run. Moreover, when the team configuring and curating a mission WGC server takes on the job of carefully preparing operational kernel sets for use by everybody else on the a project, it also eliminate the need for user applications to select individual kernels and allows instead selecting prepared kernel sets, ensuring the use of consistent data across different project functions.

B. WGC Calculation Types

Both WGC interfaces provide calculations in the following three categories: “Geometry Calculator”, “Geometry Finder”, and “Time Conversion” (Fig 4.)

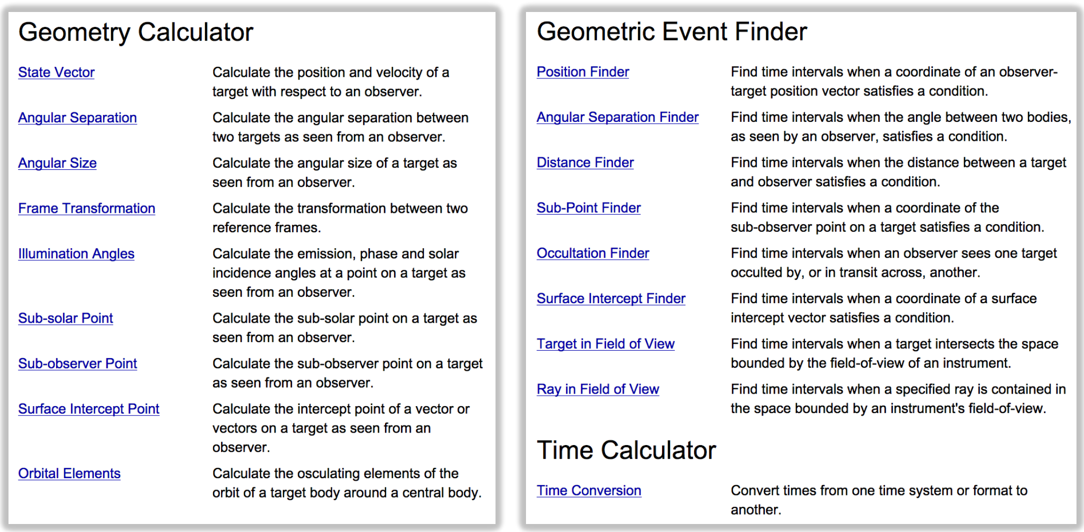

Fig. 4 WGC Calculation Types.

Calculations from the “Geometry Calculator” category compute values of various geometric parameters such as state vectors, frame transformations, and illumination angles at a single time, a set of times spaced with a fixed step within a single time interval, a set of user specified discrete times, or a set of times spaced with a fixed step within multiple intervals. These times can be specified in a variety of time systems – UTC, TDB, spacecraft on-board clock (SCLK), etc. – and formats – Calendar, Julian Date, etc. In most calculations output quantities can be computed with light time and stellar aberration corrections. In all calculations where outputs are position or velocity vectors outputs can be expressed in one of many supported coordinate systems – rectangular, planetocentric, planetographic, spherical, etc.

Calculations in the “Geometry Finder” category perform geometric event searches for time intervals when a geometric parameter has a particular value or when a geometric condition is within a single time interval or a set of multiple time intervals. In searches specifying constraints on a geometric parameter available constraints include equality, greater than, less than, in range, and local and global maxima and minima. In searches with criteria set for a particular coordinate this coordinate can be any coordinate applicable to any of the supported coordinates. In all searches the resulting intervals can be determined taking light time and stellar aberration corrections into account. Output intervals returned by any search can be altered to get their complements, be contracted or expanded, and filtered.

The calculation in the “Time Conversion” category converts times expressed in a variety of formats (calendar date, Julian date, seconds past J2000) from any of the time systems supported in SPICE (UTC, TDB, TDT, or SCLK) to another time system and format supported in SPICE. The input can be a single time, a set of times spaced with a fixed step within a single time interval, a set of discrete times, or a set of times spaced with a fixed step within multiple intervals. Where applicable the user can specify a custom output time format using the SPICE output time format specification notation.

C. WGC Graphical User Interface

When using the WGC GUI users load the WGC client page into a Web browser, select a calculation, enter required calculation inputs and click the “Calculate” button to submit a calculation request to the WGC server. Upon completing the calculation the server sends back results that are displayed below the inputs in the same browser window.

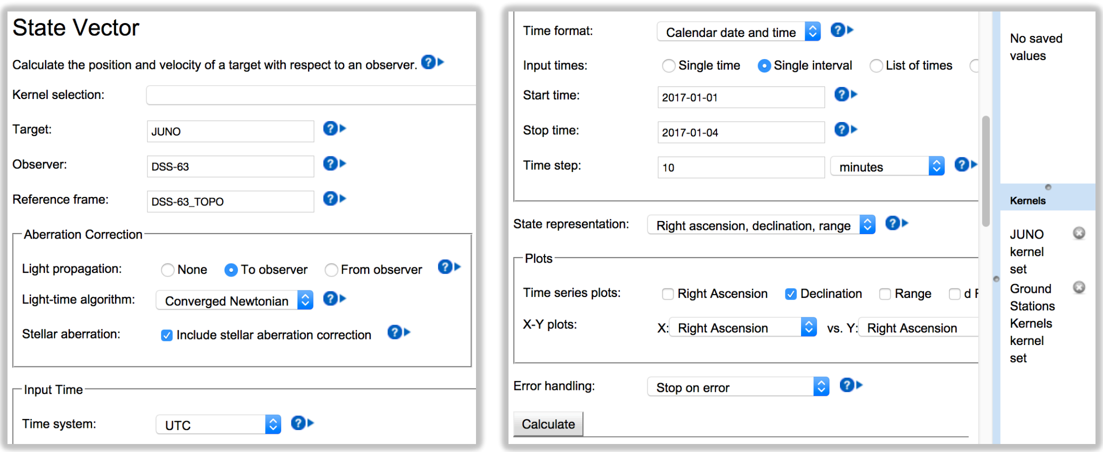

Fig. 5 WGC Inputs Example.

Fig. 5 shows inputs for a sample WGC GUI session that calculates the elevation angle of the Juno spacecraft as seen from the DSS-63 DSN tracking station in that station’s topocentic reference frame. The user uses menus and other widgets to select the two SPICE kernels sets needed to perform the calculation – the Juno mission kernel set and the ground stations kernel set, – to select the target (JUNO), observer (DSS-63), reference frame (DSS-63_TOPO) and desired aberration correction (To observer, with stellar aberration correction included), to specify the time range and step (2017-01-01 to 2017-01-04 UTC with 10 minute step) and output state’s coordinate system representation (RA/DEC), and to optionally request a time series plot for the declination (equivalent to elevation for this case). Once all required inputs are specified, the user clicks the “Calculate” button to submit this calculation request to the WGC server.

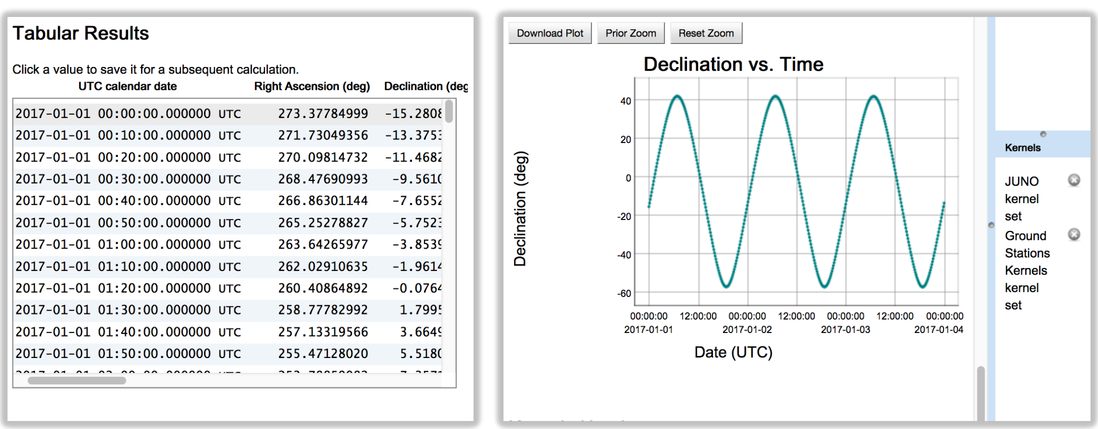

Fig. 6 WGC Outputs Example.

Fig. 6 shows the main outputs for this sample WGC calculation including the scrollable numeric outputs table and requested plot with zoom capabilities, displayed in the browser window below the inputs. In addition to the table and plot the results also include a snapshot of all inputs and the list of kernels sets and/or individual kernels used to perform the calculation. The results display also allows the user to download the numeric results in variety of formats (Excel, CSV, plain ASCII text), to download plots as PNG files, and to save individual output values and sets of "Geometry Finder" output time intervals for use in subsequent calculations.

The WGC GUI interface includes many additional features that make user interactions with the tool more efficient. These features include autocomplete for input fields specifying SPICE body and frame names based on data included in loaded kernels sets, cross-initialization of similar input fields for related calculations, and persistence of inputs and outputs when switching back and forth between calculation panels. The interface also provides calculation and field/widget specific help information and allows the maintainers of the tool to provide detailed information about kernel sets that a particular WGC server is configured to use.

D. WGC Programmatic Interface

Using the API interface (available summer 2018) user applications may access WGC calculations across the network via standard Web data access APIs. The applications, written in any language that provides built-in or third party libraries to access data via the internet, can use WGC RESTful structured URLs to submit calculation requests and to retrieve calculation results, using JSON payloads to package calculation inputs and outputs.

Calculation Request URL:

http://[host]/webgeocalc/api/calculation/new

Calculation Request POST Data:


```
{
  "calculationType": "STATE_VECTOR",
  "kernels": [
    {
      "type":  "KERNEL_SET",
      "id": 4
    },
    {
      "type":  "KERNEL_SET",
      "id": 13
    }
  ],
  "timeSystem": "UTC",
  "timeFormat": "CALENDAR",
  "intervals": [
    {
      "startTime": "2017-01-01",
      "endTime": "2017-01-04"
    }
  ],
  "timeStep": 10,
  "timeStepUnits": "MINUTES",
  "target": "JUNO",
  "observer": "DSS-63",
  "referenceFrame": "DSS-63_TOPO",
  "aberrationCorrection": "CN+S",
  "stateRepresentation": "RA_DEC"
}
```


Calculation Request Response:


```
{
  "status": "OK",
  "message": "The operation was successful.",
  "calculationId": "6ba1ade6-b7ea-4d2e-826a
    -50b6abbfa01c",
  "result": {
    "phase": "COMPLETE",
    "progress": 0
  }
}
```


Output Return URL:

http://[host]/webgeocalc/api/calculation/6ba1ade6-b7ea-4d2e-826a-50b6abbfa01c/results

Output Response:


```
{
  "status": "OK",
  "message": "The operation was successful.",
  "calculationId": "6ba1ade6-b7ea-4d2e-826a
    -50b6abbfa01c",
  "columns": [
    {
      "name": "UTC calendar date",
      "type": "DATE",
      "outputID": "DATE"
    },
    {
      "name": "Right Ascension (deg)",
      "type": "NUMBER",
      "outputID": "RIGHT_ASCENSION"
    },
    {
      "name": "Declination (deg)",
      "type": "NUMBER",
      "outputID": "DECLINATION"
    },
    ...
  ],
  "rows": [
    [
      "2017-01-01 00:00:00.000000 UTC",
      273.37784999,
      -15.28087371,
      832594791.2572775,
      -0.00276033,
      0.00317353,
      -27.30112721,
      60198.92158341,
      "2016-12-31 23:13:43.762719 UTC",
      2777.23728212
    ],
    ...
  ]
}
```


Fig. 7 WGC Programmatic Call Inputs/Outputs Example.

Fig. 7 shows example RESTful URLs and input and output JSON payloads for a WGC API calculation session that does the same calculation as the one described in the example provided in the WGC GUI section above – computing the elevation angle of the Juno spacecraft as seen from the DSS-63 DSN tracking station in that station's topocentic reference frame. The request JSON POST data include specifications of the desired calculation, two required kernel sets using their "id"s as well as all other input parameters specific to that calculation and parallel to what would be provided on the corresponding WGC GUI panel. Submitting this calculation request returns a response providing the calculation "id" and result's status and phase. The application then continues to request the calculation state using the URL "calculation/[calculationId]" until it gets a response with the phase set to "COMPLETE". At that point the application can request the calculation results using the URL "calculation/[calculationId]/results" and these results will be returned as a JSON stream closely resembling the numeric results table produced by the corresponding GUI panel, with specifications of the column names and types, and with data records containing computed parameter values.

In addition to the URLs for requesting, checking the state of, and getting the results of a calculation the WGC API interface provides URLs for retrieving the compete set of "id"s for all kernel sets available on a WGC server ("webgeocalc/api/kernel-sets") and retrieving detailed information about any of these kernels sets such as the lists of names of SPICE bodies ("kernel-set/[kernelSetId]/bodies"), frames ("kernel-set/[kernelSetId]/frames") and instruments ("kernel-set/[kernelSetId]/instruments"), data for which is included in the kernel set.

E. Operational WGC Servers

Two publicly accessible WGC servers with GUI interfaces are currently online providing SPICE-based geometry calculations for NASA and ESA planetary missions – the WGC server at NAIF, JPL and the WGC server at ESA SPICE Service, ESAC.

The WGC server at NAIF (https://naif.jpl.nasa.gov/naif/webgeocalc), in operation since 2013, is configured to work with generic multi-mission kernels sets – Solar System kernels and NASA and ESA ground station kernels, – and with archived kernel sets from the NAIF's PDS archive holdings, for missions from Viking Orbiters to Juno, 28 in total. While the archived kernel sets for on-going missions are a few months behind real-time in terms of coverage, they can be used to perform calculations for engineering and science analysis for the past. To perform geometry calculations based on the most recent predicted and reconstructed kernels users can load such kernels for many current planetary missions from the main NAIF server repository.

The WGC server at ESAC (https://www.cosmos.esa.int/web/spice/webgeocalc), a more recent installation in operation since 2017, is configured to work with archived, current operational, and future study kernel sets for the past (Rosetta, Venus Express), current (Mars Express, TGO) and future (Juice, BepiColombo) ESA planetary missions. With the current operational kernel sets automatically maintained up to date and with the future study kernel sets organized by case study scenarios, ESAC's WGC provides an excellent example of a well thought out and implemented WGC kernel set management in support of mission development and operations.

When the new WGC version with the API interface becomes available in the summer of 2018, NAIF plans to deploy it on a separate web server at JPL and to provide it to ESA SPICE Service for consideration. This new version with additional capabilities applying to both GUI and API interfaces, such as Digital Shape Kernel (DSK)-based calculations available in the latest SPICE toolkit and the ability to pre-load kernel sets at startup, is also planned be deployed on the two current WGC servers.

The NAIF group does not plan to distribute the WGC server software to the general public but might make it available to organizations involved in planetary exploration, with significant experience with SPICE and a clear need to manage their own kernel sets used by WGC. In this case NAIF will provide the WGC server binary distribution package together with the installation and kernel database configuration instructions. If interested contact the NAIF manager, Charles Acton, to discuss this possibility (Charles.Acton(at)jpl.nasa.gov).

V. SPICE-Enhanced Cosmographia

A. Introduction

To address the need for having a high-quality, freely available SPICE-based visualization application, with permission from the tool's author, NAIF took over Cosmographia, an interactive Solar System visualization program, and extended it to make use of nearly all types of SPICE kernels to accurately model observation geometry of planetary missions with extensive sets of SPICE data. With these extensions, SPICE-Enhanced Cosmographia's visualization capabilities can be applied in many areas of operations on modern planetary missions that use SPICE, from visualizing planned observation sequences to after-the-fact rendering of spacecraft positions and orientation to support engineering analysis.

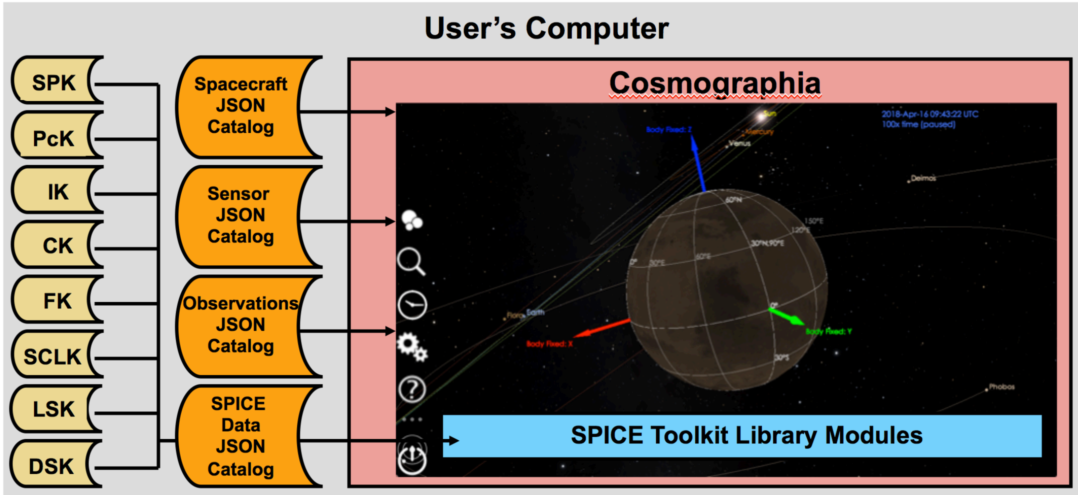

Fig. 1 SPICE-Enhanced Cosmographia Architecture.

Much like any other traditional SPICE-based application, Cosmographia is a standalone program that needs to be installed on a user workstation along with a collection of SPICE kernels required to visualize trajectories, orientations, reference frames, and instrument geometry for a space mission of interest (Fig. 8). The new objects related to that mission – spacecraft, sensors, observation sequences, and even new target natural bodies not already built into the program – are introduced into the Cosmographia "world" using catalog files in JSON format, defining these objects' properties and connecting them to SPICE as the source of their geometry information.

B. Cosmographia User Interface

Cosmographia allows users to freely move around the Solar System, placing the camera at any location and choosing camera orientation to be aligned with different rendering frames (inertial, locked, SPICE frames), to seamlessly control time flow, and to easily change general rendering characteristics (e.g. extra light) as well as the sets of objects and guides that are being shown.

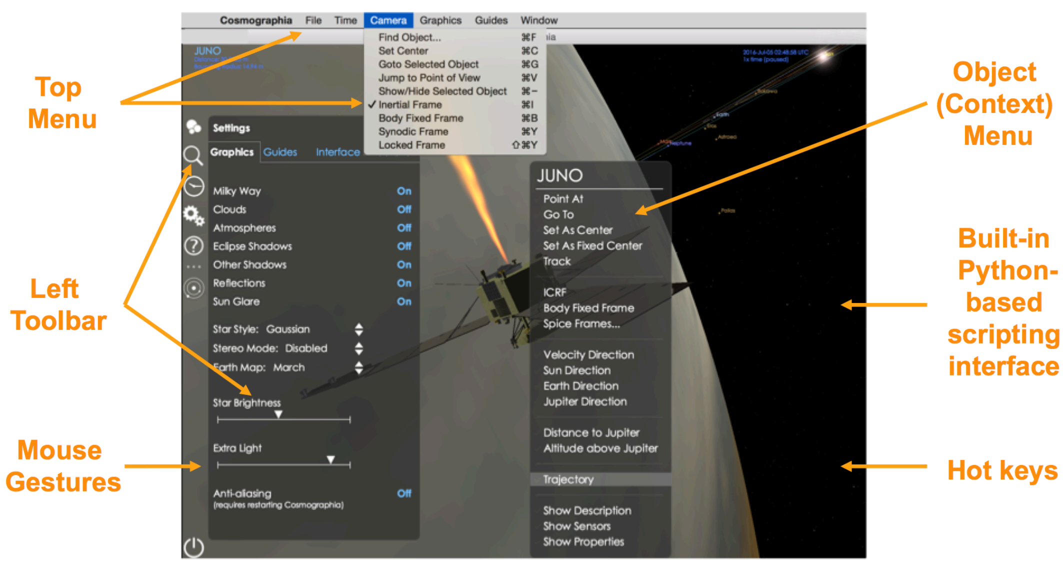

Fig. 9 Cosmographia Controls.

To achieve this flexibility the program provides many different types of controls (Fig. 9), from "Top Menu" and "Left Toolbar" options that give access to most of the program's settings and commands, to "Object (Context) Menu" items acting on and showing additional features/information for a specific object, to "Mouse Gestures" – simple drags and clicks changing the scene and moving the camera around the Solar System, to "Hot Keys" enabling quick access to many essential time, camera motion, and ancillary commands. A special place among Cosmographia controls is taken by its built-in Python based interface (available only on OS X and Linux), providing access to many of the program's elementary visualization functions and allowing users using these functions to create complex visualization sequences controlling the program without any interactive input.

C. Cosmographia SPICE-Based Features

Possibly the most important aspect of Cosmographia is that it uses SPICE to compute geometric parameters required to visualize both its built-in Solar System objects and any additional objects introduced into its "world" for mission-specific applications.

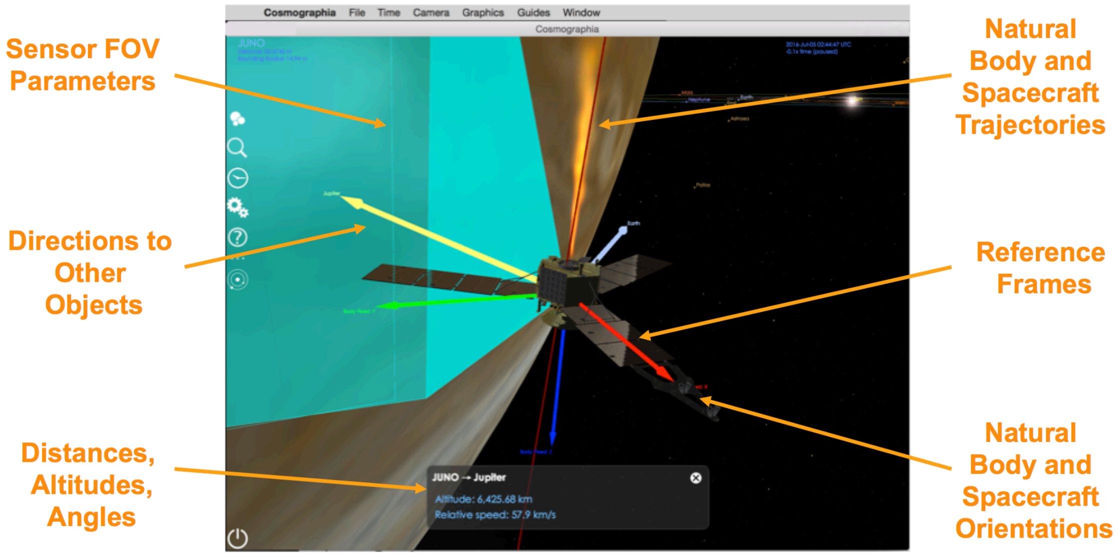

Fig. 10 Cosmographia SPICE-Based Features.

First and foremost, Cosmographia computes positions, trajectories, and orientation of all objects – the built-in Solar System bodies (planets, satellites, and main asteroids) and any spacecraft and additional natural bodies introduced into its "world" – based on SPICE ephemeris kernels (SPK) and orientation kernels (PCK, CK, FK) (Fig. 10). In addition the program reads instrument Field-Of-View (FOV) definitions needed for rendering sensor FOV cones from SPICE instrument kernels (IKs) and renders SPICE references frames based on their definitions from the frames kernels (FK) and their orientation from a variety of other SPICE kernels. The program also uses SPICE to compute directions from one object to any other object in the Solar System and to calculate a small set of numeric parameters, such as distances, altitudes, and angles, displayed in a text window at the bottom of the screen.

D. Cosmographia Mission Adaptation

To configure Cosmographia to do visualizations for a space mission, SPICE data and new objects for that mission – additional natural bodies, spacecraft, sensors, and observation sequences – are introduced into the Cosmographia "world" using catalog files in JSON format. These catalog files define the objects' properties and connect them to SPICE as the source of their geometry information.

Fig. 11 shows example SPICE data, spacecraft, and sensor catalogs for the Juno mission. The SPICE data catalog simply lists the SPICE kernels needed to compute geometry for the mission specific objects. It can list individual kernels as shown in the example or meta-kernels, which internally have to provide absolute paths to the kernels that they include. The spacecraft catalog defines the spacecraft trajectory and orientation calculation methods, spacecraft appearance (most commonly a 3D model provided in 3DS or CMOD format), and attributes of the spacecraft name label and trajectory plot. To make Cosmographia calculate the spacecraft trajectory and orientation using SPICE the spacecraft catalog simply sets the SPICE names of the target and center and the name of the SPICE reference frame associated with the spacecraft. The sensor catalog defines an instrument's FOV location and orientation, as well as its appearance attributes. To make Cosmographia use the FOV parameters from IK the catalog simply sets the instrument name attribute to the instrument SPICE name.

SPICE Data Catalog


```
{
  "version": "1.0",
  "name": "JUNO",
  "spiceKernels": [
    "jno_sclkscet_00066.tsc",
    "juno_v12.tf",
    "juno_junocam_v02.ti",
    "juno_rec_160522_160729_160909.bsp",
    "juno_rec_160729_160923_161027.bsp",
    "juno_sc_raw_160705_160705.bc",
    "juno_sc_rec_160522_160528_v01.bc",
    "juno_sc_rec_160529_160604_v01.bc",
    "juno_sc_rec_160605_160611_v01.bc",
    "juno_sc_rec_160612_160618_v01.bc",
    "juno_sc_rec_160619_160625_v01.bc",
    "juno_sc_rec_160626_160702_v01.bc",
    "juno_sc_rec_160703_160705_v01.bc",
    "juno_sc_rec_160705_160709_v01.bc",
    "juno_sc_rec_160710_160716_v01.bc",
    "juno_sc_rec_160717_160723_v01.bc",
    "juno_sc_rec_160724_160730_v01.bc",
    "juno_sc_rec_160731_160731_v01.bc",
    "juno_sc_rec_160731_160806_v01.bc",
    "juno_sc_rec_160807_160813_v01.bc",
    "juno_sc_rec_160814_160820_v01.bc",
    "juno_sc_rec_160821_160827_v01.bc",
    "juno_sc_rec_160828_160903_v01.bc",
    "juno_sc_rec_160904_160910_v01.bc",
    "juno_sc_rec_160911_160917_v01.bc",
    "juno_sc_rec_160918_160924_v01.bc"
  ]
}
```


Spacecraft Catalog


```
{
  "version": "1.0",
  "name": "JUNO",
  "items": [
    {
      "class": "spacecraft",
      "name": "JUNO",
      "startTime": "2011-08-05 UTC",
      "center": "Jupiter",
      "trajectory": {
        "type": "Spice",
        "target": "JUNO",
        "center": "Jupiter"
      },
      "bodyFrame": {
        "type": "Spice",
        "name": "JUNO_SPACECRAFT"
      },
      "geometry": {
        "type": "Mesh",
        "meshRotation": [
          0.7071, 0.7071, 0, 0
        ],
        "source": "juno.cmod"
      },
      "label": {
        "color": [ 0, 1, 1 ]
      },
      "trajectoryPlot": {
        "color": [ 0, 1, 1 ],
        "duration": "60 d",
      }
    }
  ]
}
```


Sensor Catalog


```
{
  "version": "1.0",
  "name": "JUNO",
  "items": [
    {
      "class": "sensor",
      "name": "JUNO_JUNOCAM",
      "parent": "JUNO",
      "startTime": "2011-08-05 UTC",
      "center": "JUNO",
      "trajectoryFrame": {
        "type": "BodyFixed",
        "body": "JUNO"
      },
      "geometry": {
        "type": "Spice",
        "instrName": "JUNO_JUNOCAM",
        "target": "Jupiter",
        "range": 1000000,
        "rangeTracking": true,
        "frustumColor": [
          0,
          1,
          1
        ],
        "frustumOpacity": 0.3,
        "gridOpacity": 1,
        "footprintOpacity": 0.8,
        "sideDivisions": 30,
        "onlyVisibleDuringObs": false
      }
    }
  ]
}
```


Fig. 11 Cosmographia JSON Catalogs Example.

Additional frequently used catalogs (not shown) are observation catalogs, defining times for which footprints of an instrument are to be drawn on a spheroidal target natural body, and annotation catalogs, used to display informational messages during at user specified times.

E. Cosmographia Availability and Example Operational Uses

The current SPICE-enhanced Cosmographia has been available from the NAIF server since 2015 (https://naif.jpl.nasa.gov/naif/cosmographia.html). Its OSX, Linux and Windows distribution packages include the program, all needed SPICE kernels to support computations for built-in Solar System objects, and a set of example mission adaptation catalogs for a number of missions (New Horizons, Dawn, etc.). An extensive User's Guide covering all aspects of Cosmographia use and adaptation is available at a special dedicated web site (https://cosmoguide.org).

In just two short years since its release, many planetary exploration missions at different states of development as well as some multi-mission groups chose to use SPICE-enhanced Cosmographia for various purposes. The Dawn project used it to run real-time spacecraft visualization displays in their mission support area. The Europa-Clipper and Europa Lander projects used it to produce animations of various mission periods from post launch activities to targeted Europa flybys for sequence validation and official project review purposes [3][4]. The Solar System dynamic group at JPL used it to produce many small body animations as well as in support of the radar detection for the Chandrayaan-1 spacecraft passing over the Moon's pole. The ESA SPICE Service took it a step further and provided Cosmographia adaptations for ESA missions that it supports (https://www.cosmos.esa.int/web/spice/cosmographia), allowing any team on these missions to take advantage of the program's capabilities.

VI. Conclusion

The two modern SPICE-based tools – WGC with its GUI and Web-based API interfaces, and SPICE-enhanced Cosmographia with its slick space mission visualization capabilities – are attractive alternatives to the traditional approach of using SPICE by writing custom applications linked to the SPICE toolkit. Between the two of them, they provide a simple interface to many SPICE computations, from a Web browser or via a Web API, without the need to manage kernels, and the ability to render space mission geometry situations based on accurate SPICE computations. Ongoing and future missions that use or plan to use SPICE are encouraged to consider these tools for their SPICE-based needs.

Acknowledgments

The author is pleased to acknowledge the current NAIF team members Chuck Acton, Nat Bachman and Ed Wright, and the former NAIF team members for creating the SPICE system; Mark Rose, NASA Ames, for designing and implementing WebGeocalc; Chris Laurel, Fifth Star Labs LLC, for creating the original Cosmographia, and Eric Ferguson, JPL, for enhancing it to be truly SPICE-based. The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Acton, C. H. (1996), Ancillary data services of NASA's Navigation and Ancillary Information Facility, *Planetary Space Sci.*, 44, 1, 65-70.
doi: https://doi.org/10.1016/0032-0633(95)00107-7

[2] Semenov, B. V., Acton, C. H., Bachman, N. J., Rose, M., and Wright, E. D., “WebGeocalc: Web Interface to SPICE,” *6th International Conference on Astrodynamics Tools and Techniques*, Darmstadt, Germany, 2016.

[3] Lawler, C. R., Wissler, S. S., Kulkarni, T., Ferguson E. W., and Maldague P. F., “Europa Lander Concept: High fidelity system modeling informing flight system and concept of operations years before launch,” *AIAA 15th International Conference on Space Operations (SpaceOps)*, Marseille, France, 2018 (to be published).

[4] Ferguson E. W., Wissler, S. S., Bradley, B. K., Maldague P. F., Ludwinski, J. M., and Lawler, C. R., “Improving Spacecraft Design and Operability for Europa Clipper through High-Fidelity, Mission-Level Modeling and Simulation,” *AIAA 15th International Conference on Space Operations (SpaceOps)*, Marseille, France, 2018 (to be published).