SPICE Development Plans
and Possibilities

January 2020
DSK Shape Models

Plans and Possibilities for Further Development

- **Extension of the DSK shape model subsystem**
  - Complete the Type 4 DSK code for working with digital elevation models
  - Add more functionality to the tessellated plate model (Type 2 DSK)
    » The first official release of the Type 2 subsystem, for small, irregularly shaped bodies, was released in the N66 Toolkits
  - Unfortunately NAIF has no real target date in mind for this work
• Develop SPICE 2.0: a re-implementation of the SPICE Toolkit from the ground, up, providing thread-safe and object oriented features
  – This is the major NAIF undertaking, started in May 2017
  – It is being implemented in C++11
  – It is expected to take several years

• No worries: none of the current Toolkits will be dropped.
• Continue adding capabilities to the WebGeocalc tool
  – More kinds of calculations
  – More ease-of-use features
  – This work is on-going

• Continue adding capabilities to the Cosmographia 3D mission visualization program
  – This work is on-going
Model Development

Navigation and Ancillary Information Facility

- Complete and release a large set of dynamic frames in a generic frames kernel (or kernel set)
  - Much work was done on this, but in the end it appears impossible to achieve community consensus on key aspects

- Add some aspects of ring models
  - At least ring reference frames
  - Maybe also shapes?
  - No active work on this
• Complete the Java Native Interface (JNISpice) Toolkit family
  – Reliability is felt to be very good
    » (NAIF used JNISpice to implement the WebGeocalc tool)
  – Additional documentation needs to be written

• Python interface
  – Several SPICE users have implemented and are offering their own, partial Python interfaces to SPICE
    » Check here for links to two of them
      • http://naif.jpl.nasa.gov/naif/links.html
  – NAIF’s use has been limited to preparing a few SpiceyPy lessons
  – Others report these offerings appear to be good quality products
  – Thus NAIF seems unlikely to do any of its own Python work
• 3rd parties have also implemented Ruby, Swift and Julia interfaces to CSPICE.
  – NAIF hasn’t tried testing any of these
  – NAIF does not know how complete these are
  – Give them a try, but use due caution as you do so
    » You might be able to do some one-off tests using the WebGeocalc tool as a “gold bar”
    » You could try using the “spice_discussion” bulletin board to see what other people have to say about these interfaces
Some Other Possibilities?

- More high-level SPICE 1.0 (current SPICE) computations, such as instrument footprint coverage
- More “geometry finder” computations
- Develop a more flexible and extensible instrument modeling mechanism
• NAIF is helping the Republic of South Korea learn to use SPICE in support of their upcoming Korean Pathfinder Lunar Orbiter (KPLO) mission

• Colleagues at LASP are helping the United Arab Emirates deploy SPICE in support of their upcoming Hope mission to Mars

• We hope to find the means to support upcoming science-focused SmallSat/CubeSat missions
  – Example: Lunar IceCube
What do You Suggest?

NAIF

Navigation and Ancillary Information Facility

• NAIF solicits suggestions from you!
  – How might we improve SPICE?
  – How might we improve SPICE training?
  – How might we improve NAIF’s operations?
  – How might we improve SPICE operability across the large and still growing international community?

• We’re interested in programmatic ideas as well as technical ones.