Software for Photometric and Astrometric Reduction for Video Meteors
Prakash Atreya

Date: 27 Feb, 2012

This document has been specifically prepared for The Desert Fireball Network, Imperial College. It outlines step by step instructions to setup SPICE required by SPARVM

A. Install SPICE
1. Go to http://naif.jpl.nasa.gov/naif/toolkit_IDL.html

2. Select either Linux 32 bit or Linux 64 bit version. SPARVM has not been tested in Windows, Solaris, and Mac Operating system.

3. Download icy.tar.Z and save it to your local disk. SPICE contains C binaries which are ready to use. You don't have to compile them. So you could store them anywhere you want. I usually do it under my IDLworkspaces71/icy

4. Unzip untar the SPICE file (icy.tar.Z)
   cd path_to_the_downloaded_file
   gzip -d icy.tar.Z
   tar xfv icy.tar
   rm icy.tar
   [Note: If you donot have gzip or tar, use other unzip software]

5. Readme file for icy (SPICE IDL) is at path_to_icy/doc/icy.req

6. If you still want to re-compile SPICE routines (need C compiler), you can run icy/makeall.csh

B. **Linking SPICE with IDL**
1. Find the bin directory of you IDL (for example my IDL bin path is as follows)
   set idl_bin=/usr/local/itt/idl71/bin/bin.linux.x86
   Note: make sure that the path is to the idlXX where XX is a version number.

2. Copy SPICE dlm and so files into IDL bin directories
   cp path_to_icy/lib/icy.dlm $idl_bin/.
   cp path_to_icy/lib/icy.so $idl_bin/.

3. Save test_icy.pro in your IDL default directory

4. Start IDL and compile/run test_icy.pro
Check the output of this program with the default output (also provided in the program output)

5. For details (and other options) for steps B 1- 4, see icy/doc/icy.req line 230 – 297

C. SPICE kernels

1. Download SPICE kernels from the following link:
http://naif.jpl.nasa.gov/pub/naif/generic_kernels/lsk/naif0010.tls
http://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/de421.bsp
http://naif.jpl.nasa.gov/pub/naif/generic_kernels/pck/pck00010.tpc
http://naif.jpl.nasa.gov/pub/naif/generic_kernels/pck/de-403-masses.tpc

2. Make a new folder under icy (I name it kernels) and move the above files there.

3. Save standard.ker (email attachment) (I save it under icy/kernels)
    This file is used to load all the kernels. You should know the path to the
    standard.ker file, as this is the file to be called, every time SPICE
    kernels has to be used.

    Edit standard.ker
        → Give full path to the 4 kernels downloaded above
        → Check their name (naif0010.tls.txt should be naif0010.tls)

4. Run the sample IDL program (states.pro) to check if the kernels are working.
Please modify line 124 with the path to your standard.ker.

The programs prompts user for the following. Here is an example
Enter the name of the observing body   : Earth
Enter the name of a target body   : Venus
Enter the number of states to be calculated : 3
Enter the beginning UTC time   : Thu Feb 9 12:53:29 UTC 2012
Enter the ending UTC time   : Thu Feb 16 12:53:29 UTC 2012
Enter the inertial reference frame   : J2000
Enter LT+S, LT, or NONE   : NONE

5. There are other examples you can try at icy/src/icycook/ to understand SPICE
routines

D. Install Astrolib Library

1. Download the Astrolib Library from following website or use the wget command as follows:
   Homepage : http://idlastro.gsfc.nasa.gov

   ```
   wget http://idlastro.gsfc.nasa.gov/ftp/astron.tar.gz
   ```

2. Unzip and Untar

   ```
   mkdir astrolib
   mv astron.tar.gz astrolib/.
   cd astrolib
   tar -zxvf astron.tar.gz
   ```

3. Put the Astrolib in IDL path
   If using IDLDE (this might vary depending on IDL version used)

   ```
   Click on "Window" → "Preference" → "IDL" → "Paths" → "Insert" →
   Guide it to the astrolib folder → "OK" → "Check the square box
   (this means look at the subfolder)" → "Apply" → "OK"
   ```

   If using IDL (have to do every session)

   ```
   !PATH=!PATH+':'+Expand_Path('+~/path_to_astrolib/')
   ```

4. Test some programs from Astrolib
   In IDL command line test nint.pro from astrolib which returns the nearest integer

   ```
   IDL> .compile nint.pro
   IDL> print, nint (1.4), nint (1.6)
   IDL> 1, 2
   ```