Introduction to the Family of SPICE Toolkits

March 2010

Topics

- SPICE Toolkits
  - SPICELIB (FORTRAN)
  - CSPICE (C)
  - Icy (IDL)
  - Mice (MATLAB)
- Installed Directory Structure
- Toolkit Documentation
- Toolkit Utility Programs
- Toolkit Application Programs
- Supported Platforms

SPICE Toolkit

- **The SPICE Toolkit is available in Fortran, C, IDL (Interactive Data Language), and MATLAB**
- **Toolkits contain:**
  - **Software**
    - **Subroutine libraries, with source code**
      - **SPICELIB (Fortran)**
      - **CSPICE (C)**
      - **Icy (C)**
      - **Mice (C and MATLAB script)**
    - **Executable programs**
      - **application and utility programs**
      - **cookbook examples**
    - **Installation/build scripts**
  - **Documentation**
    - **Available in ASCII and HTML**
  - **Data**
    - **Sample kernel files**
      - **Supplied ONLY for use with cookbook programs, not valid for general use**

SPICE Toolkit

- The components listed on the previous page comprise the generic Toolkit
  - Toolkits delivered to missions or other special customers may be augmented with mission- or customer-specific products
- The Fortran, C, IDL, and MATLAB Toolkits are delivered as standalone products
  - The IDL and MATLAB products include the CSPICE Toolkit
- For a given computer and operating system, the Fortran, CSPICE, IDL, and MATLAB Toolkits use identical kernel files.
    - » (See the "Porting Kernels" tutorial for information about using kernels received from a machine different from what you are using.)

Toolkit Versions

- Toolkit Version
  - Generic SPICE Toolkits have an associated Version number
    - Example: “N0063” (also written as “N63”)
  - The version number applies to the Fortran, C, IDL and MATLAB implementations for all supported platforms.
  - When does NAIF release new SPICE Toolkit versions?
    - Not according to a fixed schedule
    - Primarily driven by addition of significant new capabilities
      - For example, Icy or Mice or the geometry finder subsystem
    - On rare occasion a Toolkit update is released to fix bugs, improve documentation, or satisfy an urgent request from a flight project

Toolkit Characteristics

- Portable SPICE kernel files
- Portable NAIF Toolkit software
- Code is well tested before being released to users
- New Toolkits are always backwards compatible
  - An application that worked when linked against an older Toolkit will link and work, without need for changes, using a new Toolkit
  - Past functionality is never changed or removed
    - » Enhancements of existing routines are allowed
  - NAIF reserves the right to fix bugs
- Extensive user-oriented documentation is provided
  - Includes highly documented source code

Toolkit Library Functionality

- Toolkit libraries contain a broad set of capabilities related to the computations needed for “observation geometry” and time conversions.
  - Broad categories are enumerated on the next several pages

- Caution: not all functionality is present in all four language versions of the Toolkit library.
  - The Fortran (SPICE) and C (CSPICE) Toolkits provide virtually identical functionality.
  - The IDL (Icy) Toolkit duplicates most functionality from the C Toolkit wrapper routines.
  - The MATLAB (Mice) Toolkit provides interfaces to those routines NAIF considers the most often needed by users.
    - » Where not needed, a module is not implemented.

Toolkit Library Functionality

- Kernel read access
  - “Load” kernels
  - Get state or position vectors (SPK)
  - Get orientation of planets, natural satellites, etc. (PCK)
  - Get body shape parameters or physical constants (PCK)
  - Get orientation of spacecraft or spacecraft instruments or structures (CK, FK)
  - Get instrument parameters (e.g., FOV) (IK)
  - Query binary EK files (EK-ESQ)
- Kernel write access
  - SPK writers
  - CK writers
  - EK writers (sequence component, ESQ)
  - PCK writers (only for binary PCK files)

Toolkit Library Functionality

- **Additional ephemeris functions**
  - Classical osculating elements
  - Two-body Keplerian propagation
  - NORAD two line elements sets (TLE) propagation
  - Light time and Stellar aberration computation
- **Frame transformation**
  - Obtain 3x3 matrices for frame transformations of positions
  - Obtain 6x6 matrices for frame transformations of states
- **Time conversion**
  - Conversion between standard systems: TDB, TT (TDT), UTC
  - Conversion between SCLK and other systems
  - Parsing and formatting
- **Geometry finder**
  - Find times or time spans when a specified geometric situation is true
  - Find times or time spans when a specified geometric parameter is within a given range, or is at a maximum or minimum

Toolkit Library Functionality

- Math
  - Vector/Matrix operations
  - Rotations, Euler angles, quaternions
  - Coordinate conversion (systems: latitudinal, cylindrical, rectangular, RA and DEC, spherical, geodetic, planetographic)
  - Geometry: ellipsoids, ellipses, planes
  - High-level functions: illumination angles, sub-observer point, sub-solar point, surface intercept point.
- Constants
  - Julian date of epoch J2000, SPD(seconds per day), PI, etc.
- Strings
  - Parsing: find tokens, words
  - Numeric conversion
  - Pattern matching
  - Replace marker, substring
  - Suffix, prefix
  - Case conversion
  - Find first/last non-blank character, first/last printing character

Toolkit Library Functionality

- Arrays
  - Sorting, finding order vector, reordering
  - Searching: linear, binary
  - Insertion and deletion
- Name/code conversion
  - Bodies
  - Frames
- I/O support
  - Logical unit management (Fortran toolkits)
  - Open, read, write text files
  - Kernel pool API
- Exception handling
  - Control exception handling behavior: mode, message set, output device
  - Construct error messages

Toolkit Library Functionality

- Advanced data types
  - Cells, Sets
  - Windows (sometimes called schedules)
  - Symbol Tables
  - Planes, Ellipses

Fortran SPICE Toolkit

- SPICELIB, the Fortran SPICE Toolkit
  - Developed first: in use since February 1990
  - Contains code written in ANSI Standard Fortran 77
    - A few widely supported non-ANSI extensions are used, for example DO WHILE, DO…END DO
  - Compiles under a wide variety of Fortran compilers
    - While NAIF cannot guarantee proper functioning of SPICE under F90/F95 compilers except on officially supported environments, those compilers might properly compile SPICELIB with the resulting libraries being callable from F90/F95 code if that compiler supports the F77 standard.

CSPICE Toolkit

- **CSPICE, the C-language Toolkit**
  - Designed to duplicate the functionality of the Fortran Toolkit
  - All CSPICE source code is in ANSI C
    - » The Fortran SPICE Toolkit code is converted to ANSI C using the automatic translation program f2c
    - » High-level functions have been hand-coded in C and documented in C style in order to provide a natural C-style API. These functions are called “wrappers”
    - » Most wrappers encapsulate calls to C functions generated by f2c
      - The simpler wrappers do their work in-line to boost performance
    - » f2c’d functions may be called directly, but this is strongly discouraged since f2c’d functions emulate Fortran functionality:
      - Call by reference
      - Fortran-style array indexing
      - Fortran-style strings

continued on next page

CSPICE Toolkit

- CSPICE runs under a wide variety of ANSI C compilers
- CSPICE functions may be called from within C++ source code
  - CSPICE prototypes are protected from name mangling
- Current CSPICE Limitations
  - Not all “Required Reading” reference documents have been converted to C style, with C examples
    - Eventually all will be converted
  - CSPICE wrappers do not exist for every functionality provided in the Fortran toolkits
    - Includes all the most commonly used modules
    - More will be added as time permits
  - In some very limited cases, code generated by f2c fails to emulate Fortran accurately. Should not be a problem.
    - List-directed I/O has some problems (not consequential for CSPICE)
    - Treatment of white space in text output is slightly different in CSPICE
    - Logical unit-to-file name translation does not handle file name "synonyms" properly under Linux: once opened with a specified name, a file must be referred to using the same name throughout a program run.

Icy Toolkit

- Icy, the Interactive Data Language Toolkit
  - Provides an IDL-callable "wrapper" interface for many CSPICE wrapper routines
    - » Example:
      - CSPICE: spkezr_c (targ, et, ref, abcorr, obs, state, &ltime);
      - Icy: cspice_spkezr, targ, et, ref, abcorr, obs, state, ltime
    - » NAIF will add additional interfaces to Icy as time permits
  - By necessity the Icy Toolkit includes the complete CSPICE Toolkit.
    - » Additional Icy software components are:
      - IDL interface wrappers (implemented in ANSI C)
      - Icy cookbook programs (implemented in IDL)
  - Icy Documentation
    - » Icy Reference Guide
      - Principal documentation showing how to call Icy wrappers
      - Each Icy wrapper has an HTML page containing usage examples serving as the Icy "module header"
    - » Icy Required Reading
      - Provides background information essential for programming with Icy

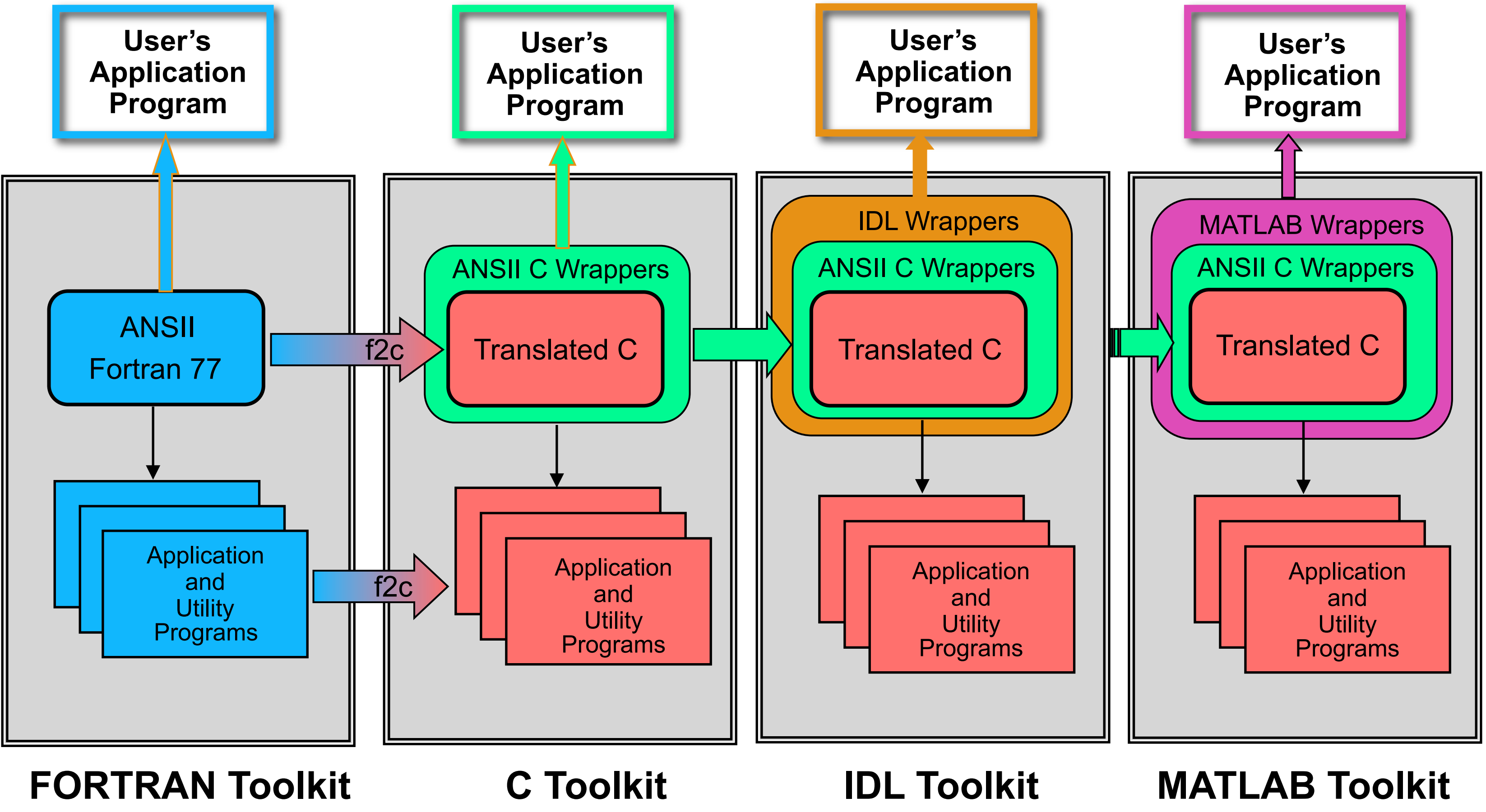
Mice Toolkit

- **Mice, the MATLAB Toolkit**
  - Mice provides a MATLAB-callable “wrapper” interface for many CSPICE wrapper routines
    - » Example:
      - CSPICE: spkezr_c (targ, et, ref, abcorr, obs, state, &ltime);
      - Mice: [state, ltime] = cspice_spkezr(targ, et, ref, abcorr, obs)
    - » More MATLAB-callable wrappers will be added as time permits
  - By necessity all Mice Toolkit packages include the complete CSPICE Toolkit.
    - » Additional Mice software components are:
      - MATLAB interface wrappers (implemented in MATLAB wrapper scripts calling the ANSI C based interface library)
      - Mice cookbook programs (implemented in MATLAB script)
  - Mice Documentation
    - » Mice Reference Guide
      - Principal documentation showing how to call Mice wrappers
      - Each Mice wrapper script has a documentation header containing usage examples, serving as SPICE “module header”, available from the `help` command. This documentation also exists as a HTML page.
    - » Mice Required Reading
      - Provides background information essential for programming with Mice

Navigation and Ancillary Information Facility

Toolkit Architecture Pictorial

- NAIF must provide SPICE capability in the popular languages
- Development and maintenance must be very economical
- Computations must be identical for all languages


User's Application Program
User's Application Program
User's Application Program
User's Application Program
IDL Wrappers
MATLAB Wrappers
ANSII C Wrappers
ANSII C Wrappers
ANSII C Wrappers
ANSII Fortran 77
f2c
Translated C
Translated C
Translated C
Application and Utility Programs
f2c
Application and Utility Programs
Application and Utility Programs
Application and Utility Programs
Interface languages →
FORTRAN Toolkit
C Toolkit
IDL Toolkit
MATLAB Toolkit

Installed Directory Structure

- The top level directory name for each Toolkit is:
  - “toolkit” for Fortran Toolkits
  - “cspice” for C Toolkits
  - “icy” for IDL Toolkits
  - “mice” for MATLAB Toolkits

- Directory structures for the Toolkits are almost identical. However…
  - The CSPICE, Icy and Mice Toolkits also have a directory for include files
  - The names for application source code directories in CSPICE, Icy and Mice differ slightly from those in the Fortran toolkit
  - Icy and Mice include additional directories for
    - » Icy/Mice source code
    - » Icy/Mice cookbook programs

Installed Directory Structure

- The next level is comprised of:
  - data
    - » Cookbook example kernels (use ONLY for training with cookbook programs)
  - doc
    - » Text documents — *.req, *.ug, spicelib.idx/cspice.idx, whats.new, dscriptn.txt, version.txt.
    - » Subdirectory containing HTML documentation, called “html”.
      - The “html” subdirectory contains a single file — the top level HTML documentation index called “index.html” — and a number of subdirectories, one for each of the various groups of documents in HTML format (API Reference Guide pages, User’s Guide pages, etc.)
  - etc
    - » In generic Toolkits this directory is empty.
  - exe
    - » Executables for brief, chronos, ckbrief, commnt, inspekt, mkspk, msopck, spacit, spkdiff, frmdiff, spkmerge, tobin, toxfr, version.
    - » Executables for the several cookbook example programs.

Installed Directory Structure

- include (applies to CSPICE, Icy, and Mice)
  - API header files.
    - File to include in callers of CSPICE is SpiceUsr.h
- lib
  - Toolkit libraries:
    - For Fortran SPICE Toolkits
      - spicelib.a or spicelib.lib (public modules; use these)
      - support.a or support.lib (private modules; don’t use these)
    - For CSPICE Toolkits
      - cspice.a or cspice.lib (public modules; use these)
      - csupport.a or csupport.lib (private modules; don’t use these)
    - For Icy Toolkits:
      - icy.so (shared object library)
      - icy.dlm (dynamically loadable module)
      - cspice.a or cspice.lib
      - csupport.a or csupport.lib
    - For Mice Toolkits:
      - mice.mex* (shared object library)
      - cspice.a or cspice.lib
      - csupport.a or csupport.lib
- src
  - Source code directories for executables and libraries
    - Files have type *.f, *.for, *.inc, *.pgm, *.c, *.h, *.x, *.pro, *.m
    - *.h files appearing here are not part of the user API

Toolkit Documentation

- All Toolkits include documentation in plain text and HTML formats
  - Plain text documents are located under the “doc” directory
  - HTML documents are located under the “<toolkit_name>/doc/html” (Unix) or “<toolkit name>\doc\html” (Windows) directory
    - » “<toolkit_name>/doc/html/index.html” or “<toolkit_name>\doc\html\index.html” is the top level index
- All Toolkits include the following kinds of documents
  - Module headers
    - » Act as primary functional specification: I/O, exceptions, particulars defining behavior of module
    - » Contain code examples
    - » A standard format is used for each routine or entry point
    - » Plain text Module Headers:
      - Fortran: the top comment block in the source code files under “src/spicelib”
      - C: the top comment block in the source code files under “src/cspice”
      - IDL: Icy Module Headers are not available in plain text format
      - MATLAB accessible via “`help` *`function_name`*” command
    - » HTML Module Headers are accessible using the “API Reference Guide” link from the top level index.

continues on next page

Toolkit Documentation

- Required Reading
  - References for principal subsystems
  - Provide many low-level details
  - Provide code examples
  - Plain text versions are located under “doc” and have extension “.req”
  - HTML versions are are accessible using the “Required Reading Documents” link from the top level index.
  - Not all of Required Readings were adapted for all languages
    - Some of the Required Reading documents provided with CSPICE still cover Fortran SPICE
    - Some of the Required Readings for Icy or Mice toolkits still cover CSPICE
- User’s Guides
  - Interface specifications for the Toolkit utility programs and applications
  - Plain text versions are located under “doc” and have extension “.ug”
  - HTML versions are accessible using the “User’s Guide Documents” link from the top level index.

Toolkit Documentation

- Other documents
  - Permuted Index
    - Maps phrases describing functionality to corresponding module names and file names
    - Shows names of all entry points in Fortran toolkit APIs
    - Plain text version is located under “doc” and has extension “.idx”:
      - Fortran: spicelib.idx
      - C: cspice.idx
      - IDL: icy.idx and cspice .idx
      - MATLAB: mice.idx and cspice.idx
    - HTML version isaccessible using the “Permuted Index” link from the top level index.
  - Toolkit Description
    - Describes the directory structure and contents of an installed Toolkit
    - Customized based on set of delivered products and platform
    - Plain text version is “doc/dscriptn.txt”
    - HTML version isaccessible using the “Toolkit Contents” link from the top level index.

Toolkit Documentation

- Other documents (continued)
  - Introduction to SPICE
    - » Brief introduction to the Toolkit and SPICE system
    - » Not available in plain text
    - » HTML version isaccessible using the “Introduction to the SPICE System” link from the top level index.
  - What’s New in SPICE
    - » Describes new features and bug fixes
    - » Plain text version is “doc/whats.new”
    - » HTML version isaccessible using the “What’s New in SPICE” link from the top level index.
  - Toolkit Version Description
    - » Indicates Toolkit version
    - » Plain text version is “doc/version.txt”
    - » Not available in HTML

Toolkit Utility Programs

- SPICE Toolkit utility programs are available to:
  - port binary SPICE kernels between incompatible systems*
    - tobin, toxfr, spacit
    - bingo (available only from the NAIF webpage)
  - port text SPICE kernels between incompatible systems
    - bingo (available only from the NAIF webpage)
  - add comments to binary kernels
    - commnt
  - read comments from binary kernels
    - commnt, spacit
    - inspekt (only for EK/ESQ files)
  - summarize coverage of binary kernels
    - brief, ckbrief, spacit
  - merge or subset SPK files
    - spkmerge
  - indicate current Toolkit version
    - version

* Usually not needed

Toolkit Application Programs

- SPICE Toolkit application programs perform various tasks:
  - create a new SPK file from a text file of state vectors or elements
    - mkspk
  - compare (diff) two SPKs
    - spkdiff
  - compare (diff) two reference frames
    - frmdiff
  - create a new CK from a text file of attitude data
    - msopck
  - carry out a wide assortment of time conversions
    - chronos
  - query Event Kernels (EKs)
    - inspekt

Supported Environments

- NAIF ports the SPICE Toolkit to several popular environments
  - Each environment is characterized by
    - » Language
    - » Hardware type (platform)
    - » Operating System
    - » Compiler
    - » Selected compilation options
- NAIF provides SPICE Toolkit packages for each supported environment
  - If you cannot find a package built for the environment of interest to you, contact NAIF
    - » Don’t try to use or port a Toolkit built for another environment

Supported Environments - Fortran

Product Name	Operating System	Compiler
Mac/Intel, OS-X, Intel FORTRAN	OS X 10.4.x	Intel Fortran 10.1
Mac/Intel, OS-X, gfortran	OS X 10.4.x	gfortran, GNU Fortran 4.3
Mac/PowerPC, OS-X, Absoft FORTRAN	OS X 10.4.x	Absoft Pro Fortran 9.0
Mac/PowerPC, OS-X, g77	OS X 10.4.x	g77, GNU Fortran 3.4.4
PC, CYGWIN, g77	Windows/Cygwin	g77, GNU Fortran 3.2
PC, Linux, Intel FORTRAN	Red Hat Linux (RHE4)	Intel Fortran 10.0
PC, Linux, g77	Red Hat Linux (RHE4)	g77, GNU Fortran 3.4
PC, Linux, gfortran	Red Hat Linux (RHE4)	gfortran, GNU Fortran 4.3
PC, Windows, Digital FORTRAN	Windows NT/2K/XP	Compaq Digital Fortran 6.0
PC, Windows, Intel FORTRAN	Windows XP	Intel Fortran 9.1
PC, Windows, Lahey FORTRAN 95	Windows NT/2K/XP	Lahey FORTRAN 95 5.6
Sun, Solaris, SUN FORTRAN	Solaris 9	Sun FORTRAN 95 8.2

Supported Environments - C

Product Name	Operating System	Compiler
Mac/Intel, OS-X, Apple C	OS X 10.4.x	gcc, GNU C 4.0.1
Mac/PowerPC, OS-X, Apple C	OS X 10.4.x	gcc, GNU C 4.0.1
PC, CYGWIN, gCC	Windows/Cygwin	gcc, GNU C 3.2
PC, Linux, gCC	Red Hat Linux (RHE4)	gcc, GNU C 3.4.6
PC, Linux, gCC, 64bit	Red Hat Linux (RHE4)	gcc, GNU C 3.4.6
PC, Windows, Microsoft Visual C	Windows NT/2K/XP	Microsoft Visual Studio .NET 7.0 C
Sun, Solaris, gCC	Solaris 9	gcc, GNU C 3.3.2
Sun, Solaris, gCC, 64bit	Solaris 9	gcc, GNU C 3.3.2
Sun, Solaris, SUN C	Solaris 9	Sun C 5.8

Supported Environments - IDL*

Product Name	Operating System	Compiler
Mac/Intel, OS-X, Apple C/IDL	OS X 10.4.x	gcc, GNU C 4.0.1
Mac/PowerPC, OS-X, Apple C/IDL	OS X 10.4.x	gcc, GNU C 4.0.1
PC, Linux, gcc/IDL	Red Hat Linux (RHE4)	gcc, GNU C 3.4.6
PC, Linux, gcc/IDL, 64bit	Red Hat Linux (RHE4)	gcc, GNU C 3.4.6
PC, Windows, Microsoft Visual C/IDL	Windows XP	Microsoft Visual Studio .NET 7.0 C
Sun, Solaris, gcc/IDL	Solaris 9	gcc, GNU C 3.3.2
Sun, Solaris, gcc/IDL, 64bit	Solaris 9	gcc, GNU C 3.3.2
Sun, Solaris, SUN C/IDL	Solaris 9	Sun C 5.8

***NAIF built and tested Icy using IDL version 6.4, but these Toolkits work with IDL 7 as well.**

Supported Environments - MATLAB*

Product Name	Operating System	Compiler
Mac/Intel, OS-X, Apple C	OS X 10.4.x	gcc, GNU C 4.0.1
Mac/PowerPC, OS-X, Apple C	OS X 10.4.x	gcc, GNU C 4.0.1
PC, Linux, gCC	Red Hat Linux (RHE4)	gcc, GNU C 3.4.6
PC, Linux, gCC, 64bit	Red Hat Linux (RHE4)	gcc, GNU C 3.4.6
PC, Windows, Microsoft Visual C	Windows XP	Microsoft Visual Studio .NET 7.0 C

***NAIF built and tested Mice using MATLAB version 7.x**