east longitude > 0 ; west longitude <0

If these predefined conventions are not convenient, you can define your own conventions by introducing the:

- solar zenith angle (degrees)
- solar azimuth angle (degrees)
- satellite zenith angle (degrees)
- satellite azimuth angle (degrees)

<u>Atmospheric Model:</u>

You can select one of the standard atmospheric profiles (taken from the LOWTRAN program)

- no gas absorption
- tropical
- midlatitude summer
- midlatitude winter
- subarctic summer
- subarctic winter
- US standard 62

If these standard profiles are not adequate, then you can select the concentration in absorbing gases and water vapor (in this case, the atmospheric model is the US standard 62):

- uw (in g cm^{-2})
- uo3 (in cm-atm)

You can also define an atmospheric profile by employing radiosondes over 34 levels.

- altitude (in km)
- pressure (in mb)
- temperature (in K)
- H_2O concentration (in g m^{-3})
- O_3 concentration (in g m^{-3})

<u>Aerosol Model:</u>

The user can choose one of the following standard aerosol models:

- no aerosols
- continental model
- maritime model
- urban model

If these standard models are not appropriate, then you can introduce your own aerosol model by defining the fraction (between 0 and 1) of each component:

- volumic % of dust-like
- volumic % of water-soluble
- volumic % of oceanic
- volumic % of soot

Aerosol concentration:

The user can fix the aerosol concentration by either entering the meteorological visibility parameter in km (the aerosol optical thickness is then calculated using a standard profile) or by directly setting the aerosol optical thickness to the 550 nm wavelength (for this option the visibility is zero).

Data defining the spectral conditions:

You can select one of the following spectral bands:

METEOSAT	vis. band	0.350	1.110
GOES EAST	vis. band	0.490	0.900
GOES WEST	vis. band	0.490	0.900
AVHRR NOAA8	1st band	0.500	0.740
	2nd band	0.690	1.080
AVHRR NOAA9	1st band	0.490	0.820
	2nd band	0.640	1.190
AVHRR NOAA10	1st band	0.540	0.780
	2nd band	0.600	1.200
AVHRR NOAA11	1st band	0.540	0.800
	2nd band	0.600	1.100
HRV SPOT	1st band	0.460	0.710
	2nd band	0.590	0.760
	3rd band	0.740	0.950
TM LANDSAT5	1st band	0.430	0.550
	2nd band	0.500	0.650
	3rd band	0.590	0.750
	4th band	0.730	0.945
	5th band	1.515	1.870
	7th band	1.975	2.405
MSS LANDSAT5	1st band	0.475	0.640
	2nd band	0.580	0.750
	3rd band	0.655	0.855
	4th band	0.785	1.100

If these conditions are not fully appropriate, you can also define your own spectral conditions using:

- a unique wavelength in micrometers, permitting a monochromatic calculation (the gas absorption is taken into account)
- the spectral boundaries (the filter function will be equal to 1 over the entire band)
- the spectral boundaries and a filter function defined using 0.005 micron intervals

Surface reflectance:

You can select a homogeneous surface, which introduces the surface reflectance, or you can define a composite structure formed of a circular target having radiance data of reflectance (r c) in an environment of reflectance (r e). The data are:

- the reflectance of the target (r c)
- the reflectance of the environment
- the radiation of the target in km

Surface reflectance (spectral variation):

The spectral dependence of the surface reflectance can be fixed according to the following options:

- a constant r value (or r c, or r e) that is independent of wavelength
- an r value (or r c, or r e) given by 0.005 micron intervals between the spectral boundaries (if the bands of a satellite are used, refer to the corresponding limits)
- the spectral reflectance of green vegetation
- the spectral reflectance of clear water
- the spectral reflectance of sand
- the spectral reflectance of lake water

Results presentation:

The SPHINX_5Sn file contains full results as well as a comparison between the simulation employing the approximate equations and a more precise numerical model (using the successive diffusion orders method). The comparison is completed for the wavelength and the geometric conditions closest to the user's, according to the following table of values:

theta s	15, 41, 60, 75 degrees
theta v	00, 10, 20, 30, 40 degrees
phi v - phi s	00, 90, 180 degrees
lambda	0.450, 0.550, 0.650, 0.850, 1.600, 2.200 microns
visibility	50, 23, 08, 05 km

The wavelength used is the wavelength equivalent to the observing spectral band. The surface reflectance used is the surface reflectance equivalent to the surface target in this spectral band.

A graphic representation is displayed showing the incident solar angle (in red) and the satellite viewing angle (in magenta) over a planisphere background. The vertical to the sub-satellite point is shown in green.

3.13 The Geometry Models Menu

Solar Zenith Angle Calculation

This function calculates the height and the solar azimuth angle for a surface point and at a given instant. The data can be supplied in either format: day/month or Julian day.

Warper

This function allows you to superimpose 512 x 512 image projections in various geometries.

The four A, B, C, and D quadrants are used as:

A: the reference image

B: the warp image

C: the zoom over reference image

D: the zoom over warp image

Before using the WARPER function, you must first place the reference image in the A quadrant and the warp image into the B quadrant.

The zooms in the C and D quadrants are designed to facilitate entering the landmarks and can be reselected with the left button from the A and B quadrants.

To begin, introduce the landmarks by clicking on SELECT MATCHING POINTS and then enter the pixel pairs by clicking the right mouse button on the reference image (quadrant A or C) and then on the warp image. In this sequence, you can also click with the left button in the A or B quadrants or move the zoom area.

If you make a mistake, the last pixel pair entered can be undone by clicking on DELETE LAST ENTRY.

After each entry, Sphinx displays:

- the number of pairs entered
- the development order of polynomials
- the standard deviation of the ensemble of landmarks

As soon as 4 landmarks have been entered, Sphinx displays in quadrant B a preview of the landmark selected in A. If the landmarks are chosen judiciously, each successive preview should confirm the process. You can also determine when you can proceed to the calculation phase.

To proceed to the calculation phase, click on SHOW RESULT (VIS. BANK) or SHOW RESULT (ALL BANKS). Once the calculation has finished, the result (image B transformed into the geometry of A) is displayed in quadrant C. Quadrant D displays the difference between A and C.

If you do not find the results satisfactory, you can add new landmarks by clicking again on SELECT MATCHING POINTS.

Note: The development order of a polynomial can be imposed by the user. The coordinates of the landmarks can be written in a file.

Note: Although the algorithm interpolates correctly, it extrapolates fairly poorly. It is therefore desirable to distribute the landmarks judiciously over the image.

Orbit Simulation

This function displays a map of the globe with lines graphing a satellite orbit on the surface. The user can modify any parameter of the satellite orbitography and redraw the new orbits.

3D Image Projection

This function constructs a 3D image using a series of 512 x 512 images. The AB quadrant or (AB,AG,AR in true color mode) contains the lower image projection (e.g., land). The BB quadrant contains the elevation of each pixel in the AB image (e.g., land model). The CB or (CB,CG,CR) quadrant contains the upper image projection (e.g., clouds) The DB quadrant contains the elevation of each pixel in the CB image (e.g., cloud altitude). The BB, CB and DB quadrants can remain empty. The user determines the viewing angles for the projection and selects the quadrant to display the 3D image.

4.0 Development of External Programs

4.1 General Principals

External programs can be used to read a Sphinx image, process the image, and then return the results to Sphinx. These programs can also pass parameters entered by the user when running the program.

Three examples of external programs are delivered with the Sphinx package. The source codes are contained in the Sphinx examples directory.

The external programs must be linked with the extlib.a library located in the Sphinx "lib" directory. WARNING: the linking must also be done with FORTRAN libraries.

4.2 Interface Functions

To communicate with Sphinx, three functions are available:

sphinx_get_para(size_x, size_y, red, green, blue, input_parameters, user_message)

input data:

char * input_parameters;

output data

char * user_message;

int size_x, size_y, red, green, blue;

This function decodes the file (tmp)sphinx_sema* and returns the image size information in lines and columns (size_x, size_y) as well as the three pointers specifying the red, green or blue planes if they are different than zero. The user_message field corresponds to editor messages that are provided by the user when using the program.

sphinx_read(input_parameters, red_image, green_image, blue_image)

input data:

char * input_parameters;

output data

unsigned char *red_image, *green_image, *blue_image;

This function reads images transmitted by Sphinx.

sphinx_write(input_parameters, red_image, green_image, blue_image)

This function returns the externally processed image back to Sphinx.

FORTRAN program example for Silicon graphics computer

```
program mire
  character *140 parameter, user_message
  logical red, green, blue
  character red_ima(1024*1024)
  character green_ima(1024*1024)
  character blue_ima(1024*1024)
  integer colonne_nb, raw_nb
c  Get external parameter
  call getarg(1, parameter)
```

```
c  Get sphinx parameters
  call sphinx_get_para(colonne_nb, raw_nb, red, green, blue,
  parameter, user_message)
c  Read sphinx images
  call sphinx_read(parameter, red_ima, green_ima, blue_ima)
c User main computations
c Send results to sphinx
  call  sphinx_write(parameter, red_ima, green_ima, blue_ima)
  stop
  end
```

<u>FORTRAN program example for Hewlett Packard computer</u>

```
program mire (parameter)
  character *140 parameter, user_message
  logical red, green, blue
  character red_ima(1024*1024)
  character green_ima(1024*1024)
  character blue_ima(1024*1024)
  integer colonne_nb, raw_nb
c  Get sphinx parameters
  call sphinx_get_para(colonne_nb, raw_nb, red, green, blue,
  parameter, user_message)
.......  identical to previous program ....
```

<u>C program example</u>

```
main(argc, argv)
    int argc;
    char * argv[];
{ char user_message[140];
  int red, green, blue, colonne_nb raw_nb, size;
  unsigned char *red_ima, *green_ima, *blue_ima;
 /* get sphinx parameters */
  sphinx_get_para(&colonne_nb, &raw_nb, &red, &green, &blue,
  argv[1], user_message, strlen(argv[1]),strlen(message));
  printf(" image size: %d X %d\n”, colonne_nb, raw_nb);
  printf(" input images red: %d green: %d blue: %d\n", red, green, blue);
  printf(" user message: %s\n", user_message);
/* memory allocations */
  size = colonne_nb * raw_nb;
  red_ima = (unsigned char *) malloc( size );
  green_ima = (unsigned char *) malloc( size );
  blue_ima = (unsigned char *) malloc( size );
```

```
/* get sphinx input data */
  sphinx_read( argv[1], red_ima, green_ima, blue_ima,
      strlen(argv[1]), size, size, size);
/* User main computation */
/* Send results to sphinx */
sphinx_write( argv[1], red_ima, green_ima, blue_ima,
      strlen(argv[1]), size, size, size);
}
```

4.3 Updating the wind_file_EXTS File

This file is located in the Sphinx files directory. It contains the list of external programs that are accessed by Sphinx. Two lines are required per program: the first line indicates the absolute file name of the executable file, and the second line is a label which is displayed under Sphinx's external functions menu.

5.0 FILES ACCESSED BY SPHINX

These files reside in the Sphinx "files" directory. They are normally under write file protection and can be neither modified nor deleted (except files as described above and below that are specifically adapted to the environment).

SPHINX_CAN_DIR	The on-line English manual
SPHINX_CFR_DIR	The on-line French manual
security	The authorization key
COMPTA	Sphinx utilization statistics
vidcolorpost	The script shell for printing to a PostScript color printer (MUST BE ADAPTED TO THE ENVIRONMENT)
vidlas	The script shell for printing to a LaserJet HP printer (MUST BE ADAPTED TO THE ENVIRONMENT)
vidlaspost	The script shell for printing to a black and white PostScript printer (MUST BE ADAPTED TO THE ENVIRONMENT)
vidpaint	The script shell for printing to a PaintJet printer (MUST BE ADAPTED TO THE ENVIRONMENT)
wind_file_EXTS	The list of external programs (MUST BE ADAPTED TO THE ENVIRONMENT)

The files below are only used by the functions for orbit simulation or satellite signal processing. They can be omitted if these functions are not utilized.

WD_MaxH	The altitudes of the globe by intervals of 1/3 of a degree
WD_PriS	The index characterizing the surfaces of the globe
coastfile.bin	The continental contours
5S_File	The constants used for the satellite signal modeling

The files below are only used by the test functions. They can be omitted if these functions are not utilized.

ima_test.R.Z	Test image
ima_test.G.Z	Test image
ima_test.B.Z	Test image

6.0 Files Created by Sphinx

The location of temporary files (tmp) can be directed to any directory. If you do not have enough space in /usr/tmp, you can launch Sphinx using the option -U dir, with dir being the directory name in which Sphinx places its temporary files.

Example: Sphinx -U /WORK/tmp

A list of files or temporary files produced by Sphinx follows.

($HOME/).SP_ALGEBRA	Saves the equations (IMAGE ALGEBRA menu)
($HOME/).sphinx_fmt	User specific image description
(tmp/)sphinx_masked	Temporary file used to memorize the masked planes
(tmp/)sphinx_crt0	Temporary file
(tmp/)sphiNx_crt0	Temporary file
(tmp/)sphiMx_crt0	Temporary file
(tmp/)sphinxG_crt0	Temporary file
(tmp/)sphinx_data*	Temporary files for communicating
(tmp/)sphinx_sema*	with the external programs
(./)clsize.rs	CLUSTER ANALYSIS results
(./)SPHINX_CLAn	PIXEL CLASSIFICATION results
(./)SPHINX_FT	Saves FOURIER function
(./)SPHINX_GJETn	Printing of LaserJet gray scale images
(./)SPHINX_GRAPHX	Saves graphics
(./)SPHINXGRAXYn	Extracted data from graphs
(./)SPHINX_LJETn	Printing of LaserJet graphics
(./)SPHINX_PCAn	PRINCIPAL COMPONENTS ANALYSIS results
(./)SPHINX_PJETn	Printing of PaintJet color images
(./)SPHINX_PSn	Printing of PostScript graphics
(./)SPHINX_PSCn	Printing of PostScript color images
(./)SPHINX_PSGRn	Printing of PostScript gray scale images
(./)SPHINX_VALn	PIXEL VALUES results
(./)SPHINX_WARPERn	WARPER results
(./)SPHGRAXYn	EXTRACT VALUES results in REDRAW SAVED GRAPH
(./)SPHINX_5Sn	SATELLITE SIGNAL SIMULATION results
(./)SP_ALGEBn	IMAGE ALGEBRA real results
(./)Save_ContV.R (.G .B)	Save session files
(./)Save_Cont.CM	Save session files
(./)Save_ContM.R (.G .B)	Save session files

7.0 Setting Up and Running Sphinx

7.1 Installation

Installing Sphinx requires about 8 megabytes of disk space. To install Sphinx:

- create a Sphinx installation directory (/usr/sphinx for instance) and use this directory as the default.

- read the distribution tape: tar xv

- modify the Sphinx startup script: bin/sphinx (this script contains the necessary information for the adaptation).

- modify the printing scripts for the printer configuration of your network. These scripts are the files vidlas, vidlaspost, vidcolorpost and vidpaint.

The Sphinx directory contains the following subdirectories:

bin	the executables
examples	the external program examples
files	the files used by Sphinx
lib	the extlib.a library for the external programs
util	the utilities

The contents of the files directory is described in chapter 6.

7.2 Startup

To launch Sphinx simply type the command sphinx. The startup takes about one minute, after which a small window appears showing you how to call up the main Sphinx window.

In the Sphinx demonstration version the functions for writing files and printing files are deactivated (in the menus, the windows corresponding to these functions appear in gray).

Sphinx is compatible with the following window managers: uwm, mwm, twm and olwm (Sun openwin).

In the event of installation and startup problems with Sphinx, consult below the paragraph entitled "Trouble Shooting Sphinx".

After the installation you can immediately start working with Sphinx by using the READ IMAGE TEST function (in the TEST menu). This function reads a test image delivered with Sphinx. To display the image in true color mode, you must first use the TRUE COLOR function in the COLOR menu.

WARNING: Remember to place the sphinx/bin directory in the PATH of the users before startup.

WARNING: The X server must be started before Sphinx, and note that Sphinx must be started from an XWindows window.

WARNING: If you operate Sphinx under twm or openwin, the startup script bin/sphinx must be edited to add the -G option to the startup of sphinx.out. With this option, Sphinx connects to the X server in order to access the X color tables.

WARNING: Various Sphinx functions terminate execution with an error if you do not have access to the default directory or if the default directory already contains Sphinx files that were created by another user.

7.3 Trouble Shooting Sphinx

Begin by verifying that the installation modifications mentioned above have been correctly followed.

Symptom	Not enough disk space to install Sphinx.
Remedy	Certain supplementary files delivered with Sphinx can be suppressed: see chapter 6.
Symptom	Sphinx does not launch.
Remedy	Verify that your PATH includes the Sphinx bin directory.
Symptom	Sphinx starts and then stops after displaying the message "Cannot open DISPLAY"
Remedy	Verify the variable shell DISPLAY value (echo $DISPLAY). It should be set to either unix:0.0 or host:0.0 (host being the machine name of which Sphinx uses the screen). Verify that the X11 server (or X11/NeWS on Sun) is active. Verify that the server accepts the connections made from your machine (commands xhost or xauth on Sun).
Symptom	Incorrect color display.
Remedy	Start Sphinx using the -G option.
Symptom	Sphinx indicates that it is in demonstration mode, and the functions for file writing and printing are inactive.
Remedy	Consult Mélodie.
Symptom	An error message from the X server indicates a color allocation problem. This situation arises on certain HP stations as a result of the X server configuration.
Remedy	See the file: /usr/lib/X11/X0screens.
Symptom	Insufficient disk space in the /usr/tmp directory.
Remedy	Start up Sphinx using the -U dir option (dir being the name of a directory that Sphinx uses in place of /usr/tmp for storing temporary files).
Symptom	On the READ IMAGE command Sphinx hangs. Lack of disk space in /tmp.
Remedy	Increase the available disk space for /tmp. One solution can be to place /tmp on a machine having more free disk space. Another solution is to make a symbolic link of /tmp to another directory (for instance, rmdir /tmp and then ln -s /usr/tmp /tmp).
Symptom	The print command produces nothing.
Remedy	Verify that the scripts files/vidlaspost, files/vidlas, files/vidpaint, files/vidcolorpost are properly installed.
Symptom	The X server displays color allocation error.
Remedy	This problem occurs on DecStations when the X server has been started in true color mode. To solve the problem, start up the X server in PseudoColor, which is done by adding -class PseudoColor after /usr/bin/Xtm on the file line /etc/ttys beginning

with 0: (for the modification to take affect, you must quit and restart the system).

Symptom	The external programs do not run.
Remedy	Verify that the file files/wind_file_EXTS are properly installed. Delete the (/tmp)/sphinx_data* (/tmp)/sphinx_sema* files.