Subject: MODIS43: Question on processing albedo lookup table...
Posted by kim20026 on Thu, 10 May 2007 16:17:17 GMT
View Forum Message <> Reply to Message

Hi, everyone! I am back to ask a question on albedo processing with IDL.

I am dealing with MODIS43 for processing albedo. As most of you may know, however, MODIS43 only provides black sky albedo (BSA) and white sky albedo (WSA). We need to calculate actual albedo based on these values.

Recently I downloaded the look up table (sky_lut.dat) and several source codes from ...
http://www-modis.bu.edu/brdf/userguide/tools.html

However, these files are written in C. Because I am not familiar with C yet, I got stuck now.

At this point, I have two questions...

1. Is there anyone who has been working on MODIS43-albedo processing with IDL (not with C)? If there is someone, please let me know.

2. I found the 'spawn' statement dealing with the codes written in other computer languages (i.e: FORTRAN or C) in IDL environment. But I don't know how to begin. If I want to run 'tell_skyl.c' code with 'spawn' statement, what can I do?

Harry

Tell_skyl.c

```
/**
 * Calculate fraction of diffuse skylight through a look-up table
 * which was pre-calculated with the 6S code for seven MODIS land
spectral
 * bands plus three broadbands (visible, NIR and shortwave)
 * [-od]    aerosol optical depth at 550nm [0..1]
 * [-szn]   solar zenith angle [0..89]
 *
 * by Feng Gao at Boston University on July 2002
 * contact fgao@bu.edu
 *
 **/

#include <stdio.h>
#include <math.h>
```

```
typedef struct
{
  int aerosol_type;
  int bandno;
  float solar_zenith;
  float optical_depth;
  float skyl;
} SKYL;

void read_skyl_table (float skyl_lut[2][10][90][50]);
float get_skyl (SKYL s,float skyl_lut[2][10][90][50]);

void main(int argc, char **argv)
{

  int i,band;
  float skyl_lut[2][10][90][50];
  SKYL s;

  if(argc!=5) {
    printf("Usage example: %s -od 0.2 -szn 30.0\n", argv[0]);
    exit(1);
  }

  /* parse command line */
  for(i=1;i<argc;i++){
    if(strcmp(argv[i],"-od")==0)
      s.optical_depth=atof(argv[++i]);
    else
      if(strcmp(argv[i],"-szn")==0)
 s.solar_zenith=atof(argv[++i]);
      else{
 printf("\nWrong option:%s\n",argv[i]);
 printf("Usage example: %s -od 0.2 -szn 30.0\n", argv[0]);
 exit(1);
      }
  }

  /* read lookup table from input file "skyl_lut.dat" */
  read_skyl_table(skyl_lut);

  printf("BAND    Fraction_of_diffuse\n");
  for(band=0; band <10; band++) {
    s.bandno=band;
    /* assume continental type, but finally need to get it from
BRDF_type
       0: continental 1: maritime */
```

```
    s.aerosol_type=0;
    /* get SKYL from SKYL lookup table (it depends on optical depth,
       solar zenith angle, aerosol type, and bands) */
    s.skyl=get_skyl(s,skyl_lut);
    printf(" %2d          %5.3f\n",band+1,s.skyl);
  }
}


/*******************************************************
   read SKYL LUT from pre-generated LUT file (skyl_lut.dat)
   2 aerosol types
   10 bands (7 MODIS bands + 3 broad bands (VIS, NIR, SW)
   90 degrees (0-89 degrees with 1 degree step
   50 optical depth ( 0-1 with 0.02 step)
*******************************************************/
void read_skyl_table (float skyl_lut[2][10][90][50])
{

  char str[200];
  FILE *in;
  int aerosol,band,szn,od,NBANDS;

  NBANDS=10;  /* MODIS defaults */

  if((in=fopen("skyl_lut.dat","r"))==NULL) {
    printf("Can't open SKYL LUT file (skyl_lut.dat) \n");
    exit(1);
  }

  for(aerosol=0;aerosol<2;aerosol++) {
    fscanf(in,"%s %s\n",str,str);
      for(band=0;band<NBANDS;band++) {
 fscanf(in,"%s %s\n",str,str);
 for(od=0;od<51;od++)
   fscanf(in,"%s ",str);
 for(szn=0;szn<90;szn++){
   fscanf(in,"%s ",str);
   for(od=0;od<50;od++) {
     fscanf(in,"%f ",&(skyl_lut[aerosol][band][szn][od]));
   }
 }
     }
  }
  fclose(in);
}

/* get SKYL value from LUT */
```

```
float get_skyl (SKYL s,float skyl_lut[2][10][90][50])
{
  int szn,od;

  szn=(int)(s.solar_zenith+0.5); /*get solar zenith index */
  if(szn==90) szn=89;
  od=(int)(s.optical_depth/0.02+0.5); /*get optical depth index */
  if(od>=50) od=49;
  return skyl_lut[s.aerosol_type][s.bandno][szn][od];
}
```