Subject: Re: Sunrise/Sunset Algorithm
Posted by hwljn on Sun, 12 Feb 2017 08:33:08 GMT
View Forum Message <> Reply to Message

> Here is realization of sunrise-sunset alg. I did it almost year ago
> I checked it with example data and I checked it with my own data and data from http://www.weather.com/weather for some places. The errors were less then 5 min.
>
> Change the lat,lon,day,month,year,zenith,offset for your conditions
>
> ;Main Procedure
> PRO SUNRISETIME
> ;for testing use http://www.weather.com/weather
> ;
>
> ;control input
> lat=40.9
> lon=-74.3
> day=25
> month=6
> year=1990
> zenith=90.83
> offset=-4
> riseTime=getSunriseTime(LATITUDE=lat,LONGITUDE=lon,DAY=day,MONTH=month,YEAR=year,ZENITH=zenith)
> pSunriseTime=OBJ_NEW('IDLTIMECLASS')
> pSunriseTime->setDay,day
> pSunriseTime->setMonth,month
> pSunriseTime->setYear,year
> IF (riseTime EQ -1) THEN RETURN
> riseTime=pSunriseTime->convertToLocalTime(riseTime,offset)
> pSunriseTime->parseHoursOfDay,riseTime
> pSunriseTime->outputTimestamp;();
>
> print,"----------"
>
> sunsetTime=getSunsetTime(LATITUDE=lat,LONGITUDE=lon,DAY=day,MONTH=month,YEAR=year,ZENITH=zenith)
> pSunsetTime=OBJ_NEW('IDLTIMECLASS')
> pSunsetTime->setDay,day
> pSunsetTime->setMonth,month
> pSunsetTime->setYear,year
> IF (sunsetTime EQ -1) THEN RETURN
> sunsetTime=pSunriseTime->convertToLocalTime(sunsetTime,offset)
> pSunsetTime->parseHoursOfDay,sunsetTime
> pSunsetTime->outputTimestamp;()
>

```
>    ;destroy objects
>    OBJ_DESTROY,pSunriseTime,pSunsetTime
>
>  END
>
>  ;Calculus sunrise time for location
>  ;inputs latitude, longitude,day,month,year,zenith
>
>  FUNCTION getSunRiseTime,
LATITUDE=lat,LONGITUDE=lon,DAY=day,MONTH=month,YEAR=year,ZEN ITH=zenith
>    PRINT,"Sunrise Calculations"
>    N1=FLOOR(275*month/9)
>    N2=FLOOR((month+9)/12)
>    N3=(1+FLOOR((year-4*FLOOR(year/4)+2)/3))
>    N=N1-(N2*N3)+day-30
>    lngHour=lon/15
>    t=N+((6-lngHour)/24)
>    M=(0.9856*t)-3.289
>     L=M+(1.916*SIN(!CONST.DtoR*M))+(0.020*sin(2*!CONST.DtoR*M))+ 282.634
>    IF (L LT 0.) THEN L=L+360
>    IF (L GT 360.) THEN L=L-360
>    RA=!CONST.RtoD*atan(0.91764*tan(!CONST.DtoR*L))
>    Lquadrant=(FLOOR(L/90))*90
>    RAquadrant=(FLOOR(RA/90))*90
>    RA=RA+(Lquadrant-RAquadrant)
>    RA=RA/15
>    sinDec=0.39782*sin(!CONST.DtoR*L)
>    cosDec=cos(asin(sinDec));;
>     cosH=(cos(!CONST.DtoR*zenith)-(sinDec*sin(!CONST.DtoR*lat)))
/(cosDec*cos(!CONST.DtoR*lat))
>    IF (cosH GT 1.) THEN BEGIN
>      PRINT,"The Sun is never rises (for this date) !!!"
>      UT=-1
>    ENDIF ELSE BEGIN
>      H=360-(!CONST.RtoD*acos(cosH))
>      H=H/15
>      T1=H+RA-(0.06571*t)-6.622
>      UT=T1-lngHour;
>      IF (UT LT 0) THEN UT=UT+24
>      IF (UT GT 24) THEN UT=UT-24
>      PRINT,"Sunrise time (UT) is ",UT
>    ENDELSE
>    RETURN, UT
>  END
>
>  ;Calculus sunset time for location
>  ;inputs latitude, longitude,day,month,year,zenith
>
```

```
>  FUNCTION getSunSetTime,
LATITUDE=lat,LONGITUDE=lon,DAY=day,MONTH=month,YEAR=year,ZEN ITH=zenith
>    PRINT,"Sunset Calculations"
>    N1=FLOOR(275*month/9)
>    N2=FLOOR((month+9)/12)
>    N3=(1+FLOOR((year-4*FLOOR(year/4)+2)/3))
>    N=N1-(N2*N3)+day-30
>    lngHour=lon/15
>    t=N+((18-lngHour)/24)
>    M=(0.9856*t)-3.289
>     L=M+(1.916*SIN(!CONST.DtoR*M))+(0.020*sin(2*!CONST.DtoR*M))+ 282.634
>    IF (L LT 0.) THEN L=L+360
>    IF (L GT 360.) THEN L=L-360
>    RA=!CONST.RtoD*atan(0.91764*tan(!CONST.DtoR*L))
>    Lquadrant=FLOOR(L/90)*90
>    RAquadrant=FLOOR(RA/90)*90
>    RA=RA+(Lquadrant-RAquadrant)
>    RA=RA/15
>    sinDec=0.39782*sin(!CONST.DtoR*L)
>    cosDec=cos(asin(sinDec))
>     cosH=(cos(!CONST.DtoR*zenith)-(sinDec*sin(!CONST.DtoR*lat)))
/(cosDec*cos(!CONST.DtoR*lat))
>    IF (cosH LT -1.) THEN BEGIN
>      PRINT,"The Sun is never sets (for this date) !!!"
>      UT=-1
>    ENDIF ELSE BEGIN
>      H=!CONST.RtoD*acos(cosH)
>      H=H/15
>      T1=H+RA-(0.06571*t)-6.622
>      UT=T1-lngHour;
>      IF (UT LT 0) THEN UT=UT+24
>      IF (UT GT 24) THEN UT=UT-24
>      PRINT,"Sunset time (UT) is ",UT
>    ENDELSE
>    RETURN, UT
>  END
>
>  ;class for time
>
>  ;declaration
>  PRO IDLtimeCLASS__DEFINE
>     A={IDLtimeCLASS,day:0,month:0,year:0,hours:0.,minutes:0.,sec onds:0.}
>  END
>
>  ;constructor
>  FUNCTION IDLtimeCLASS::INIT
>    RETURN, 1
>  END
```

```
>
>  ;set day
>  PRO IDLtimeCLASS::setDay,day
>    SELF.day=day
>  END
>
>  ;set month
>  PRO IDLtimeCLASS::setMonth,month
>    SELF.month=month
>  END
>
>  ;set year
>  PRO IDLtimeCLASS::setYear,year
>    SELF.year=year
>  END
>
>  ;set hours
>  PRO IDLtimeCLASS::setHours,hours
>    SELF.hours=hours
>  END
>
>  ;set minutes
>  PRO IDLtimeCLASS::setMinutes,minutes
>    SELF.minutes=minutes
>  END
>
>  ;set seconds
>  PRO IDLtimeCLASS::setSeconds,seconds
>    SELF.seconds=seconds
>  END
>
>  ;get day
>  FUNCTION IDLtimeCLASS::getDay
>    RETURN,SELF.day
>  END
>
>  ;get month
>  FUNCTION IDLtimeCLASS::getMonth
>    RETURN,SELF.month
>  END
>
>  ;get year
>  FUNCTION IDLtimeCLASS::getYear
>    RETURN,SELF.year
>  END
>
>  ;get hours
>  FUNCTION IDLtimeCLASS::getHours
```

```
>    RETURN,SELF.hours
>  END
>
>  ;get minutes
>  FUNCTION IDLtimeCLASS::getMinutes
>    RETURN,SELF.minutes
>  END
>
>  ;get seconds
>  FUNCTION IDLtimeCLASS::setSeconds
>    RETURN,SELF.seconds
>  END
>
>  ;parse the hours of day (hh.hh -> hh mm ss.ssss)
>  PRO IDLtimeCLASS::parseHoursOfDay,time
>    hours=FIX(time)
>    tmpTime=time-hours
>    minutes=FIX(tmpTime*60)
>    seconds=(tmpTime*60-minutes)*60
>    SELF.hours=hours
>    SELF.minutes=minutes
>    SELF.seconds=seconds
>  END
>
>  ;output timestamp
>  PRO IDLtimeCLASS::outputTimestamp
>     timestampString=TIMESTAMP(YEAR=SELF.year,MONTH=SELF.month,DA
Y=SELF.day,HOUR=SELF.hours,$
>    MINUTE=SELF.minutes,SECOND=SELF.seconds)
>    PRINT,timestampString
>  END
>
>  ;update for localtime
>  FUNCTION IDLtimeCLASS::convertToLocalTime,time,offset
>    time=time+offset
>    IF (time LE 0) THEN time=24+time
>    RETURN,time
>  END

YOUR JOB IS SO WELL!
THANKS A LOT!
```