

时间转换--年积日-儒略日-GPS 周

一、年积日、儒略日、GPS 周等简介

儒略历

公元前 46 年，罗马执政官儒略·凯撒颁布了儒略历。儒略历每年 12 个月，平年 365 天，闰年 366 天。除 2 月外，单数月份 31 天，偶数月份 30 天。2 月份平年 29 天，闰年 30 天。每隔 3 年置一闰年。

儒略历是纯太阳历，每年的平均长度是 365.25 天。每 400 年，儒略历与回归年大约相差 3 天，即： $(365.25-365.242189)*400 = 3.1244$ 凯撒为了纪念改历成功，将他出生的 7 月从 Quintilis 改成自己的名字 Julius

儒略日和简化儒略日

在天文学有一种连续纪日的儒略日（JD），它以儒略历公元前 4713 年 1 月 1 日的 GMT 正午为第 0 日的开始。还有一种简化儒略日（MJD）： $MJD=JD-2400000.5$

MJD 的第 0 日是从公历 1858 年 11 月 17 日的 GMT 零时开始的。我写完前一个句号时的 MJD 是 53583.22260。小数部分是以 UTC 时间在当天逝去的秒数除以 86400 得到的。0.22260 约为 UTC 时间的 5:20，加上中国的时区就是 13:20。

年积日

年积日是仅在一年中使用的连续计时法。每年的 1 月 1 日计为第 1 日，2 月 1 日为第 32 日。平年的 12 月 31 日为第 365 日，闰年的 12 月 31 日为第 366 日。

格里高利历——公历

公元 1582 年 3 月 1 日，罗马教皇格里高利十三世颁布了格里高利历，将不能被 4 整除的世纪年算作平年，这就是我们现在使用的公历。

在公历中，每 400 年有 97 个闰年，平均每年的长度是：

$$(365 \times 400 + 97) / 400 = 365.2425 \text{ 天}$$

每 400 年，公历与回归年大约相差 0.1244 天，即：

$$(365.2425 - 365.242189) \times 400 = 0.1244$$

GPS 时、GPS 周、周内秒

GPS 周（GPS Week）是 GPS 系统内部所采用的时间系统。

时间零点定义的为：1980 年 1 月 5 日夜晚与 1980 年 1 月 6 日凌晨之间 0 点。最大时间单位是周（一周：604800 秒）。每 1024 周（即 7168 天）为一循环周期。第一个 GPS 周循环点为 1999 年 8 月 22 日 0 时 0 分 0 秒。即从这一刻起，周数重新从 0 开始算起。星期记数规则是：Sunday 为 0，Monday 为 1，以此类推，依次记作 0~6，GPS 周记数（GPS Week Number）为“GPS 周 星期记数”。

表示方法：从 1980 年 1 月 6 日 0 时开始起算的周数加上周内时间的秒数（从每周周六/周日之夜开始起算的秒数，例如：1980 年 1 月 6 日 0 时 0 分 0 秒的 GPS 周：第 0 周，第 0 秒。

例如：2004 年 5 月 1 日 10 时 5 分 15 秒的 GPS 周：第 1268 周，第 554715 秒，GPS 周记数（GPS Week Number）为 1268 6，第 554715 秒。

三、程序实现成果(具体代码见附件)

1. gps2cal 将 GPS 周和周内秒转换到公历时间

```
function cal=gps2cal(gpst)
% gps2cal 将 GPS 周和周内的秒转换到公历 GPS 时间
% cal=gps2cal(week, sec) 返回的公历是 1x6 矩阵，6 列分别为年月日时分秒
% gpst: 1x2 矩阵，2 列分别为 GPS 周和周内的秒

% GPS 从 MJD44244 开始
mjd=44244+(gpst(1)*86400*7+gpst(2))/86400;
cal=mjd2cal(mjd);
```

2. gps2cal1 由公历日期和 GPS 周内秒计算公历时间

```
function cal=gps2cal1(date, tow)
% gps2cal1 由公历日期和 gps 周内秒计算公历 GPS 时间
% cal=gps2cal1(date, tow) 返回的公历是 1x6 矩阵，6 列分别为年月日时分秒
% date: 1x3 矩阵，3 列分别为公历年月日
% tow: GPS 周内秒

mjd=cal2mjd(date);

% GPS 从 MJD44244 开始
week=floor((mjd-44244)/7);
cal=gps2cal([week, tow]);
```

3. cal2gps 将公历 GPS 时间转换到 GPS 周和周内秒

```
function gpst=cal2gps(cal)
% cal2gps 将公历 GPS 时间转换到 GPS 周和周内的秒。
% gpst==cal2gps(cal) 返回的 gpst 是 1x2 矩阵，2 列分别为 GPS 周和周内秒
% cal: 1x6 矩阵，6 列分别为年月日时分秒。构造 cal 时可以省略末尾的 0

if length(cal) < 6
    cal(6)=0;
end
mjd=cal2mjd(cal);
% GPS 从 MJD44244 开始
elapse=mjd-44244;
week=floor(elapse/7);
elapse=elapse-week*7; % 周内天数
gpst=[week round(elapse*86400)];
```

4. mjd2cal 将简化儒略日转换到公历时间

```
function cal=mjd2cal(mjd)
% mjd2cal 将简化儒略日转换到公历年月日时分秒。
% cal=mjd2cal(mjd) 返回的 cal 是 1x6 矩阵，6 列分别为年月日时分秒
% mjd: 简化儒略日

jd=mjd+2400000.5;
cal=jd2cal(jd);
```

5. cal2mjd 将公历时间转换到简化儒略日

```
function mjd=cal2mjd(theyear,themoth,theday);
%return the julian day (year,day)定义函数 julia 为转换为 julian 日期函数,变量 y 为返回的 julian 日期,输入变量为 theyear/年,themoth/月, theday/日
leapyear=rem(theyear,4); %年对 4 求余数
if leapyear>0
    leap=0; %判断余数大于 0, 不能被 4 整除, 则用 leap 标记这一年不是闰年 (标记为 0)
else
    if rem(theyear,100)==0 & rem(theyear,400)~=0
        leap=0; %能被 4 整除但是不能被 400 整除也不是闰年, 则用 leap 标记这一年不是闰年
```

```

else
    leap=1;          %其他情况是闰年
end
end
%%%%%%采用平年的日历，按 1-12 月分月计算 julian 日期
if themonth==1
    juliaday=theday;
end
if themonth==2
    juliaday=theday+31;
end
if themonth==3
    juliaday=theday+59;
end
if themonth==4
    juliaday=theday+90;
end
if themonth==5
    juliaday=theday+120;
end
if themonth==6
    juliaday=theday+151;
end
if themonth==7
    juliaday=theday+181;
end
if themonth==8
    juliaday=theday+212;
end
if themonth==9
    juliaday=theday+243;
end
if themonth==10
    juliaday=theday+273;
end
if themonth==11
    juliaday=theday+304;
end
if themonth==12
    juliaday=theday+334;
end
if leap==1
    if themonth<=2      %如果是闰年，当月份小于 2 月时，julian 日期与平年相同
        juliaday=juliaday;
    end
end

```

```

    end
    if themonth>=3
        juliaday=juliaday+1;    %当月份大于 2 月时，在平年的基础上加一天
    end
end
mjd(1)=theyear;
mjd(2)=juliaday;              %返回的矢量 y，第一个元素为年，第二个元素为 Julian
日

```

6. jd2cal 从儒略日计算公历时间

```

function cal=jd2cal(J)
% 从儒略日计算公历年月日时分秒
% cal=jd2cal(J)
% 返回的 cal 是 1x6 矩阵，6 列分别为年月日时分秒
%
% 公元 1582 年 10 月 4 日 24:00 点之前使用儒略历，公元 1582 年 10 月 15 日 00:00 点之后
使用公历

if (J < 1721423.5) % 公元 1 月 1 日 0 时
    BC = 1;
else
    BC = 0;
end

%start from Julian March 1, 4801 B.C.

if( J < 2299160.5 ) % before 1582.10.4. 24:00 is Julian calender
    j0=floor(J+0.5);
    dd=J+0.5-j0;
else % after 1582.10.15. 00:00 is Gregorian calender
    %number of certury years that are not leap year
    n1=floor((J-2342031.5)/36524.25/4)+1;    %1700.3.1.0
    n2=floor((J-2378555.5)/36524.25/4)+1;    %1800.3.1.0
    n3=floor((J-2415079.5)/36524.25/4)+1;    %1900.3.1.0
    j0=n1+n2+n3+J+10;
    dd=j0+0.5-floor(j0+0.5);
    j0=floor(j0+0.5);
end

```

```

j0=j0+32083;
year0=ceil(j0/365.25)-1;
year=year0-4800;
day=j0-floor(year0*365.25);
month=floor((day-0.6)/30.6)+3;
day=day-round((month-3)*30.6);
if month>12
    month=month-12;
    year=year+1;
end

year=year-BC;

sec=round(dd*86400);

hour=floor(sec/3600);
sec=sec-hour*3600;
min=floor(sec/60);
sec=sec-min*60;
cal=[year, month, day, hour, min, sec];

```

7. cal2jd 将公历时间转换到儒略日

```

function jd=cal2jd(cal)
% cal2jd1 将公历年月日时分秒转换到儒略日。
% jd=cal2jd(cal) 返回儒略日
% cal: 1x6 矩阵, 6 列分别为年月日时分秒。构造 cal 时可以省略末尾的 0
%
% 公元 1582 年 10 月 4 日 24:00 点之前使用儒略历, 公元 1582 年 10 月 15 日 00:00 点之后
% 使用公历

if length(cal) < 6
    cal(6)=0;
end
year=cal(1);
month=cal(2);
day=cal(3)+(cal(4)*3600+cal(5)*60+cal(6))/86400;

y = year + 4800; %4801 B.C. is a century year and also a leap year.

if( year < 0 )
    y =y+ 1;    % Please note that there is no year 0 A.D.
end

m=month;

```

```

if( m <= 2 )    % January and February come after December.
    m = m+12;
    y = y - 1;
end

e=floor(30.6 * (m+1));

a=floor(y/100); % number of centuries

% 教皇格雷戈里十三世于 1582 年 2 月 24 日以教皇训令颁布，将 1582 年 10 月 5 日至 14 抹
% 掉。1582 年 10 月 4 日过完后第二天是 10 月 15 日
if( year <1582 )|(year==1582&month<10)|(year==1582& month==10 &day<15)
    b = -38;
else
    b = floor((a/4) - a); % number of century years that are not leap years
end

c=floor(365.25* y); % Julian calendar years and leap years

jd= b + c + e + day - 32167.5;

```

8. cal2wd 借助 MJD，由公历年月日推算星期几，按照格里高利十三世的历法改革去掉 1582 年 10 月 5 日至 14 日

```

function wd=cal2wd(cal)
% cal2wd    将公历年月日转换到星期几
% wd=cal2wd(cal) 返回星期几
% cal: 1x3 矩阵，3 列分别为年月日
%
% 这个函数根据儒略日推算星期。
% 根据教皇格雷戈里十三世的历法改革，儒略日中没有 1582 年 10 月 5 日至 14 日。
% 1582 年 10 月 4 日过完后第二天是 10 月 15 日。
% 这个函数在公元 1582 年 10 月 4 日 24:00 点之前使用儒略历，公元 1582 年 10 月 15 日 00:00
% 点之后使用公历

mjd=floor(cal2mjd(cal));

% 2009 年 3 月 9 日 (MJD 54899) 是星期一
wd=mod(mjd-54899, 7)+1;

```


9. cal2wd1 由公历年月日推算星期几，按照英国人的做法去掉 1752 年 9 月 3 日至 13 日 cal2wd2 不借助 MJD，由公历年月日推算星期几，去掉 1582 年 10 月 5 日至 14 日

```
function wd=cal2wd1(cal)
% cal2wd 将公历年月日转换到星期几
% wd=cal2wd1(cal) 返回星期几
% cal: 1x3 矩阵，3 列分别为年月日
%
% 英国人在 1752 年才采用格利戈里历法。英国人去掉了 1752 年 9 月 3 日至 13 日。
% 这个函数在公元 1752 年 9 月 2 日 24:00 点之前使用儒略历，公元 1752 年 9 月 14 日 00:00
% 点之后使用公历

Y=cal(1);
M=cal(2);
D=cal(3);

if ((M==1)||(M==2)) % 一月、二月当作前一年的十三、十四月
    M=M+12;
    Y=Y-1;
end

if ((Y<1752)||((Y==1752)&&(M<9))||((Y==1752)&&(M==9)&&(D<3))) % 判断是否在
1752 年 9 月 3 日前
    A=mod((D+2*M+floor((3*(M+1))/5)+Y+floor(Y/4)+5),7); % 1752 年 9 月 3 日前的公
式
else
    A=mod((D+2*M+floor((3*(M+1))/5)+Y+floor(Y/4)-floor(Y/100)+floor(Y/400)),7);
    % 1752 年 9 月 3 日后的公式
end

wd=A+1;
```