정확성 1~2분내의 행성위치계산법

By Paul Schlyter, Stockholm, Sweden
email: pausch@saaf.se or paul.schlyter@ausys.se
WWW: http://welcome.to/pausch or http://hotel04.ausys.se/pausch

번역 및 편집 : 지용호(池龍浩)
전자우편 : jiyh78@hanmail.net
홈페이지 : http://www.astronote.org

오늘날 행성들의 궤도 자료들로부터 행성들의 위치를 계산하는 것은 어렵지 않습니다. 단지 필요한 것은 컴퓨터와 적당한 프로그램만 있으면 됩니다. 만약 그런 프로그램을 제작하고 싶다면 이 페이지에서 필요한 식들을 제공받을 수 있을 것입니다. 이 페이지에서 목표는 20세기에서 21세기 동안에 행성들의 위치를 1내지 2분(2/60도) 오차 이내로 행성들의 위치를 얻어내는 것이고 더불어 소행성및 혜성들의 위치도 그들의 궤도 자료들로부터 구해낼 것입니다.

다양한 종류의 컴퓨터와 계산기가 서로 다른 언어를 이용하기 때문에 여러 종류의 프로그램 언어를 다룰 것입니다. 하나의 프로그램언어로부터 다른 언어로 번역하는 것보다는 구미에 맞는 언어로 주어지는 식들을 변환하는 것이 프로그래밍을 하는데는 수월할 것이기 때문이죠. 그러므로 여기선 프로그램 소스 뿐 아니라 프로그래밍 방법 자체를 제공하는데 치중하겠습니다. 각 식들은 한 수치적인 예제이므로 이 식을 프로그램에 적용할 때 수월할 것입니다. 한 가지 기억해야 할 것은 여기서 나오는 모든 수치적 값들은 약간의 오차가 있다는 것입니다. 그러므로 여기서 계산하는 결과와 실제값은 약간 다르게 나올 것입니다. 이 장에서 계산되어지는 값들은 전부 1-2분이내의 범위안의 오차만 허용할 것입니다.

참고로 여기서 쓰는 변수들은 그리스 문자는 쓰지 않을 것입니다. 적경은 λ,적위는 δ를 쓰는데 대신해 각각 RA, Decl을 쓸 것입니다. 그 이유는 여기서는 응용하는 분야이고 컴퓨터 언어가 그리스문자는 인식하지 못하기 때문입니다. 이곳에서의 대부분의 자료는 스웨덴의 Paul Schlyter와 Stockholm가 쓴 Computing Planetary Positions 자료를 해석, 정리한 것임을 알립니다.

정확성에 대한 몇가지 언급 A few words about accuracy

여기서는 미리 말했듯이 위치를 1내지 2분(1분=1/60도)이내로 계산한다고 했었습니다. 이 정도의 정확성을 위해 되도록이면 가장 간단하게 계산하는 방법을 쓸 것입니다. 실제계산에 많은 오차를 내지 않는한 추가되야할 항목을 생략해 나갈 것입니다. 다음에 설명되는 것은 우리가 목표로한 정확성에 맞추기 위해 가정한 사실들을 열거한 것입니다.

1. 장동(章動 nutation : 자전하는 물체의 회전축의 세차운동(歲差運動)에 수반하여 일어나는 단주기의 진동운동)과 광행차(光行差 aberration : 관측자의 운동에 의해서 천체의 겉보기위치(位置)가 영향을 받는 현상)는 무시합니다.

2. 행성간의 수차(즉 빛의 여행 시간)을 무시했습니다.

3. 역표시(曆表示 ephemeris time : 지구의 공전에 근거를 둔 시계(時系))와 자전시(自轉時 terrestial time : 지구의 자전에 근거를 둔 시계) 및 그리니치평균태양시(UT)간의 차이를 무시합니다.

4. 세차(歲差, precession : 적도면과 황도면의 교점인 춘분점이 황도를 따라 2만 5800년을 주기로 천천히 서쪽으로 이동하는 현상)를 황도좌표계의 경도에 단순히 더하는 것으로 계산하였습니다.
5. 행성궤도요소의 고계도항은 무시하였습니다. 이것은 지금으로부터 1000년안에 2분을 넘기는 오차를 발생시길 것입니다. 달의 경우 지금으로부터 1000년 뒤에 약 7분정도의 오차가 발생할 것이다. 이 오차는 지금으로 부터 시간에 제곱함으로서 더욱 큰 오차를 낼 것입니다.

6. 대부분의 행성간에 섭동을 무시했습니다. 단지 주요 섭동을 일으키는 달, 목성, 토성, 천왕성은 포함시켰습니다. 만약 정확성을 요구하지 않는다면 섭동은 무시해도 될 것입니다.

7. 가장 큰 섭동을 가지는 해왕성-천왕성간의 섭동은 이들의 궤도요소에서 설명됩니다. 그러므로 해왕성과 천왕성의 정확성은 더 줄어들 것입니다. 특별히 먼 미래와 과거에는 우리가 원하는 오차범위내로 구할 수 없습니다. 그러므로 계산시에는 가까운 세기에 해당되는 시간만 입력해야 할 것입니다.

기본적인 지식 Fundamentals

행성이나 태양, 달의 위치를 계산하는데 필요한 지식에 관해 논해보도록 하겠습니다. 여기에 나오는 용어는 반드시 알아야 하는 것이고 이해가 되었다면 다음으로 넘어가셔도 무방합니다. 각각의 용어는 숙지해 주시고 이해해 주시기 바랍니다.

여기서는 지구에서 행성이나 달, 태양을 관측할 때 그들의 위치를 표시하는 방법에 관해서 논합니다. 더불어 궤도요소들에 관해서 논하게 될 것입니다.

천체(행성, 위성등)는 아시다시피 항상 태양주위를 타원궤도로 돕니다. 그러나 정확한 타원궤도가 아니라 다른 행성들 간의 섭동으로 인해 이 타원궤도에서 약간 이탈하게 됩니다. 그냥 간단히 위치를 계산할려는 사람은 이 섭동부분을 제거해 그냥 어림치만 구할 수도 있을 것입니다. 그렇지만 처음에 우리는 위치계산의 오차를 1-2분으로 줄이려고 했기 때문에 섭동부분은 반드시 집고 넘어가야 하는 부분입니다.

행성들의 궤도의 특징을 간단히 살펴보면 금성이나 해왕성은 원궤도에 매우 가깝습니다. 이들과 달리 수성과 명왕성은 원궤도에서 많이 이탈되어 있습니다. 즉, 이심률이 다른 행성들보다 크다고 할 수 있습니다. 대부분 소행성은 평균적으로 이보다 더 큰 이심률을 가집니다. 또 혜성들이 소행성들보다 더 큰 이심율을 가집니다. 예를들어 헬리혜성은 근일점이 금성보다도 가깝고 원일점은 해왕성보다도 멉니다. 어떤 혜성들은 평균적으로 포물선궤도에 가까운 이심률이 아주 큰 궤도를 가지기도 합니다. 포물선 궤도는 궤도 자체가 닫혀있지 않기 때문에 이러한 궤도를 가진 혜성은 태양을 한번 지나가고 다시는 돌아오지 않습니다. 포물선 궤도는 타원궤도를 거의 무한적으로 늘려 놓은 것과 거의 같으므로 이런 궤도를 가진 혜성(거의 무한적으로 늘려놓은 타원궤도)은 수십만년 뒤에 다시 돌아올 것입니다.

이제 실제적으로 기본지식에 관해 논하겠습니다.

원일점과 근일점 - aphelion and perihelion
행성이 태양에 각각 가장 멀때와 가까울 때 궤도의 한 지점을 말합니다.
단 포물선 궤도는 단지 근일점(perihemium)만 가집니다.

원지점과 근지점 - apogee and perigee
달 또는 지구를 도는 인공위성이 지구에 각각 가장 멀때와 가까울 때 궤도의 한 지점을 말합니다.

천구 - the celestial sphere
관측자 주위의 임의의 거리를 가진 가상적인 구입니다.

천구의 적도 - the celestial equator
지구의 적도를 천구에 투영한 것입니다.

천구의 황도 - the ecliptic
지구의 궤도면. 이것은 또한 지구에서 바라볼때 태양이 일년동안 겉보기 움직임의 궤적입니다. 황도는 천구의 적도로 부터 약 23.5도 기울어져 있습니다. 황도는 천구의 적도를 두점(춘분점-the Vernal Point,추분점-the Autumnal Point)을 가로지릅니다. 춘분점은 흔히 적도좌표계(equatorial coordinates)와 황도좌표계(ecliptic coordinates)가 달리 사용되어지는 기준점이라 할 수 있습니다..

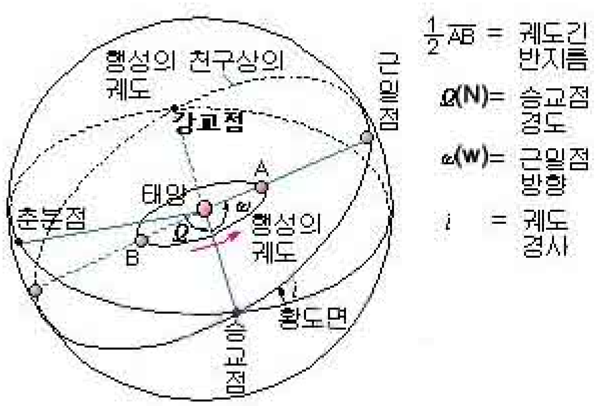
적경과 적위 - Right Ascension and Declination

이 값들은 **천구의 적도를 기준면**으로 사용하는 **적도좌표값**입니다. 춘분점에서 적경과 적위는 모두 값이 0입니다. 적경은 **춘분점(기준점)**을 기준으로 천구의 **적도**를 따라 태양이 가는 방향으로 측정되며 항상 시간과 분으로 측정됩니다. (1시간은 15도이다) 적위는 천구의 적도를 기준으로 북쪽을 양(+)로 남쪽을 음(-)으로 지정해 범위는 =90도에서 -90도로 지정됩니다.

황경과 황위 - Ecliptic longitude and Ecliptic latitude
이 값들은 **천구의 황도를 기준면**으로 사용하는 **황도좌표값**입니다. 춘분점에서 황경와 황위는 모두 값이 0입니다. 황경는 **춘분점(기준점)**을 기준으로 천구의 **황도**를 따라 태양이 가는 방향으로 측정되며 그 값은 항상 도(degrees)으로 측정됩니다. 황위는 천구의 황도를 기준으로 북쪽을 양으로 잡습니다.

태양중심, 지구중심, 지구의 지표면 중심 - Heliocentric, Geocentric, Topocentric
태양중심과 관련된 위치는 태양중심 , 지구의 중심과 관련된 위치로 지구중심, 지구의 표면과 관련되어 있는 위치는 지구의 지표면 중심으로 측정됩니다. 지표면과 지구중심의 차이는 달을 제외한 나머지 천체들을 1내지 2분이내의 오차값을 감안하였으므로 무시해도 좋습니다. (몇몇 경우에 따라서 소행성은 지구에 매우 가깝게 접근하므로 지표면에서의 위치를 생각해야 할 것입니다.)

궤도 6요소 - The orbital elements



이 요소들은 일종에 각 천체들의 궤도의 위치를 나타내주는 기본적인 **6가지** 값으로 **장반경, 이심율, 근일점 통과시간, 근일점의 위치, 궤도경사, 승교점의 황위(또는 경도)**로 구성되며 천구상에서 그 천체의 시간에 따른 원, 타원, 포물선, 쌍곡선 궤도의 위치를 완전하게 정의할 수 있습니다. 그 중 3가지 값은 궤도의 크기 및 모양을 나타내며, 나머지 3가지 값은 행성의 위치를 나타냅니다. 꼭 숙지하시기 바랍니다.

a : 평균거리 또는 장반경 - semi-major axis
e : 이심율 - eccentricity
T : 근일점 통과 시간

원궤도는 이심율(e)가 0입니다. 타원은 이심율이 0에서 1 사이이고 포물선은 이심율이 1입니다. 마지막으로 쌍곡선은 1보다 큰 값을 가집니다. 포물선은 무한 장반경을 가집니다. 그러므로 근일점의 거리만 다음과 같이 주어질 수 있습니다.
q : 근일점 거리 = a * (1 - e)
관습적으로 포물선과 가까운 이심율을 가진 쌍곡선 궤도와 타원궤도도 대신해 q로 주어집니다.

나머지 3가지 요소는 공간상에서 궤도의 방향을 정의합니다.

i : 기울기 - inclination
즉, 황도와 관련된 궤도의 기울기(tilt)를 말합니다. 기울기는 0에서 180도의 값을 가집니다. 만약 기울기가 90도보다 크면 행성은 역행을 하게 되는 것입니다. 즉 태양이 가는 반대방향으로 돌게 됩니다. 역행궤도를 가지는 잘 알려진 천체는 헬리혜성입니다.
N : (보통 "Omega:Ω" 문자로 쓰여집니다.) 승교점(천체가 황도면(기준면)을 기준으로 남반구에서 북반구로 지나갈때 천체의 궤도와 만나는 점. 즉, -위도에서 +위도로 넘어가는 찰라의 점)의 황

경. 이 각은 춘분점으로부터 황도를 따라 승교점까지 각도를 말합니다.

w : (보통 "Small Omega:ω" 문자로 쓰여집니다.) 승교점으로부터 근일점까지의 천체의 궤도를 따라 잰 각.

위의 6개 값은 주궤도 요소들입니다. 이것은 뒤에 나오는 부궤도 요소값들로부터 계산되어질수 있습니다.
q : 근일점거리 - perihelion distance = a * (1 - e)

Q : 원일점거리 - aphelion distance = a * (1 + e)

P : 궤도의 주기 - Orbital period = 365.256898326 * a**1.5/sqrt(1+m) days
여기서 m은 태양의 질량을 1로 했을시 천체의 상대적 질량(혜성이나 소행성은 이 값을 0으로 주어도 무방). sqrt()는 제곱근 함수입니다.

n : 일일움직임 - Daily motion = 360_deg / P degrees/days

t : 현재 율리우스 일(julian day)을 말합니다..

T : 천체가 근일점에 있을 때 율리우스 일을 말합니다.

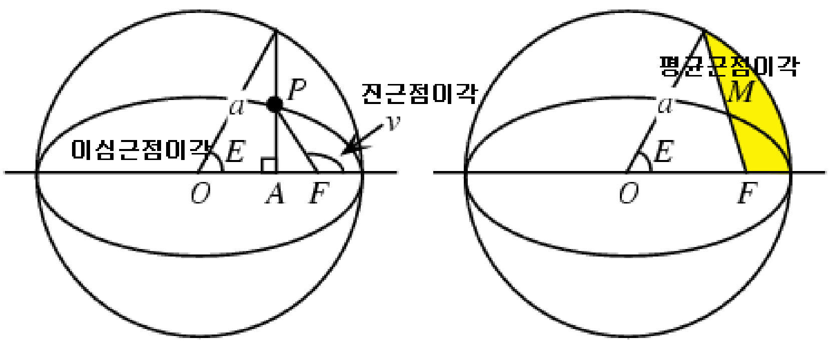
t - T : 근일점이후로의 적일(積日)을 말합니다.

M : 평균근점이각(平均近點離角) - Mean Anomaly = n * (t - T) = (t - T) * 360도 / P
* 근점각이라고도 하는 근점이각(近點離角, Anomaly)은 행성과 같이 타원궤도 위를 움직이는 천체가 근일점으로부터 떨어져 있는 각도를 말합니다. 특히 2pi(360도)를 공전 주기로 나눈 것을 평균가속도라고 하고, 이러한 움직임을 평균운동이라 하며, 평균운동을 한다고 가정한 천체의 진근점이각(眞近點離角;아래참조)을 평균근점이각이라고 합니다. 모든 근점이각(평균근점이각, 진근점이각, 이심근점이각)은 근일점과 원일점에서 각각 0도와 180도로 어느것이나 같습니다.

L : 평균경도 - Mean Longitude = M + w + N
평균근점이각에 승교점의 경도와 근일점의 위치각을 더한 값입니다

E : 이심근점이각(離心近點離角) - Eccentric Anomaly
이심근점이각은 케플러 방정식에서 M = E - e * sin(E)로 정의됩니다. 이것은 타원궤도에 위치를 계산하기위한 보조 각도입니다.

v : 진근점이각(眞近點離角) - True Anomaly
태양에서 바라볼때 근일점에서 행성까지의 각도를 말합니다.



근 점 이 각

r : 태양중심에서의 거리 - Heliocentric distance
태양에서 행성까지의 거리

다음 관계는 타원궤도로부터 확실히 유도되어질 수 있습니다

x = r * cos(v) = a * (cod(E) - e)
y = r * sin(v) = a * sqrt(1 - e*e) * sin(E)
x,y,z 직각좌표계의 값. 태양중심에서 바라볼때 위치는 지구중심에서 볼때의 위치를 변환할 때 직각좌표값을 사용합니다.

다음에는 행성위치를 계산할때 사용되는 몇가지 함수들을 정의하고 이런 함수를 프로그래밍은 어떻게 해나가는지에 대해서 알아보겠습니다.

몇몇 유용한 함수들 Some useful functions

앞에서와 같은 궤도 계산을 할때 몇몇 유용한 함수로 접근하는 것이 유용합니다. 그 함수중 프로그래밍 언어의 표준 라이브러리에 있지만 없는 것들은 프로그래머가 더 첨가해야 합니다. 전자계산기 같은 경우 라디안(radian, 호도(弧度))값과 도(degree, 도(度))값에서 sin/cos/tan 함수와 그의 역함수를 계산할 수 있습니다. 어떤 계산기는 직각좌표와 극좌표 사이를 직접적으로 변환해주는 함수를 가지는 것도 있습니다.

개인용 컴퓨터 같은 경우 계산만을 전문적으로 하는 전자계산기보다는 이러한 함수를 전부지원해주지는 못합니다. 프로그래밍 언어 BASIC에서는 단지 sin/cos/tan와 atan(=arctan)를 라디안값범위 안에서만 제공합니다. arcsin 과 arccos은 제공하지 않으며 그리고 도값에서 제공되어진 삼각함수를 사용할 수 없습니다. 그러나 다음 예에서와 같이 제공되지 않은 함수를 기존에 있는 함수를 이용해서 만들어 낼 수는 있습니다.

```
 95 rem Constants
100 pi = 3.14159265359
110 radeg = 180/pi

115 rem arcsin and arccos
120 def fnasin(x) = atn(x/sqr(1-x*x))
130 def fnacos(x) = pi/2 - fnasin(x)

135 rem Trig. functions in degrees
140 def fnsind(x) = sin(x/radeg)
150 def fncosd(x) = cos(x/radeg)
160 def fntand(x) = tan(x/radeg)
170 def fnasind(x) = radeg*atn(x/sqr(1-x*x))
180 def fnacosd(x) = 180 - fnasind(x)
190 def fnatand(x) = radeg*atn(x)

195 rem arctan in all four quadrants
200 def fnatan2(y,x) = atn(y/x) - pi*(x<0)
210 def fnatan2d(y,x) = radeg*atn(y/x) - 180*(x<0)

215 rem Normalize an angle between 0 and 360 degrees
217 rem Use Double Precision, if possible
220 def fnrev#(x#) = x# - int(x#/360#)*360#

225 rem Cube Root (needed for parabolic orbits)
230 def fncbrt(x) = exp( log(x)/3 )
```

위에 코드들은 전통적인 MS BASIC에서 쓰여질 수 있습니다. 만약 다른 BASIC 종류를 사용한다면 이 코드들은 조금 수정되어져야만 합니다.
위에 코드들은 우리에게 라디안과 도값에 대해서 sin/cos/tan와 그의 역함수를 제공합니다. 위에 주어지는 함수중에 fnatan2()와 fnatan2d()은 두 매개변수 x,y를 가지고 arctan(y/x)값을 구해냅니

다. 그러나 이 함수는 -90도 < arctan(y/x) < 90도 범위 안에서만 구해집니다. 그러므로 범위를 -180도에서 180도로 구해낼려면 이 값에 -180 * (x<0)을 더해주어야 합니다. 여기서 x<0이 true 일때 -1, false일때 0을 돌려줍니다. 이 함수들을 이용해 각도를 구하면 이것은 직각좌표에서 극좌표로 변환부분에 쓰여집니다. 이 함수의 구동원리는 밑에 표로 주어져 있습니다. 참고로 거리는 다음과 같이 구해집니다.

distance = sqrt(x*x + y*y)

fnatan2d(y,x)의 작동원리			
개요 : fnatnd(y/x)는 -90도 < f(y,x) < 90도만 구한다. -180도 < f(y,x) < 180도를 구하기 위해 fnatnd(y/x)에 -180도 * (x<0)을 더해준다.(x<0이 true일때 -1, false일때 0을 반환)			
구 분	fnatand(y/x)	x<0	fnatan2d(y/x)
1사분면	0 < f(y,x) < 90도	0(false)	0 < f(y,x) <90도
2사분면	-90도 < f(y,x) < 0도	-1(true)	90도 < f(y,x) < 180도
3사분면	0 < f(y,x) < 90도	-1(true)	180도 < f(y,x) < 270도
4사분면	-90도 < f(y,x) < 0도	0(false)	-90도 < f(y,x) < 0도

fnrev# 함수는 모든 각도를 0에서 360도 사이로 변화시켭니다. 여기서 #은 함수와 변수가 실수형을 쓴다는 의미입니다. 이 함수를 정수형보다는 실수형으로 쓰여집니다. 이 함수에 대해선 다음에 다루도록 하겠습니다.

주의사항 : 이들 함수중 몇몇은 0으로 나누어 질 수 있을 것입니다. 만약 fnasin(1.0)을 계산하면 그것은 atn(1/sqrt(1-1.0*1.0)) = atn(1/sqrt(0)) = atn(1/0)으로 계산되어져 큰 문제가 생길 수 있습니다. 이렇게 1/0을 BASIC로 계산하면 "Overflow"와 같은 에러메세지가 나옵니다. 실제로 이와 같이 정확히 1을 대입하는 것은 아주 드문일입니다. arctan에 가능한 가장 큰 숫자를 대입해서 계산할 때 거의 pi/2(또는 90도)가 구해집니다. 이것은 자명한 결과입니다. 그러나 만약 더 많은 내부함수를 가지는 더욱더 현대적인 구조를 가진 BASIC를 사용한다면 그때는 arcsin의 더 좋은 버젼을 사용하여 특별한 경우로써 arcsin(1.0)을 취급할 수 있을 것입니다.

유사한 문제점으로 단지 양수만 계산할 수 있는 fncbrt()함수가 있습니다. 예를들어 fncbrt(0)이면 log(0)이 되어 수학기본법칙에 어긋나는 경우가 발생합니다. 이런 경우도 마찬가지로 많은 내장함수 정의와 함께 음수와 0과 같은 값을 가지고 계산할 수 있게 다시 정의할 수 있습니다. 만약 한가지 간단한 규칙을 들자면 0의 세제곱근은 물론 0이고 음수의 세제곱근은 음수를 양수로 만들어 계산합니다. 그리하여 바뀌어진 양수로 세제곱근을 계산하고 결과는 허수로 만드는 것입니다.

다음 두가지 대중적인 프로그래밍 언어인 C와 C++의 표준라이브러리에는 BASIC보다 더 많은 종류의 삼각함수가 갖추어져 있습니다. 이들 언어에는 sin/cos/tan와 그들의 역함수가 주어지며 이들 함수 중에는 atan2()함수도 있습니다. 여기서 우리가 해주어야 하는 것은 도값에서 삼각함수를 얻기위한 몇몇 매크로를 정의하고 0과 360도 사이로 만들어주는 rev()함수를 정의하는 것입니다. 다음으로 세제곱근을 계산하는 cbrt()도 정의하겠습니다.

```
#define PI          3.14159265358979323846
#define RADEG       (180.0/PI)
#define DEGRAD      (PI/180.0)
#define sind(x)     sin((x)*DEGRAD)
#define cosd(x)     cos((x)*DEGRAD)
#define tand(x)     tan((x)*DEGRAD)
#define asind(x)    (RADEG*asin(x))
#define acosd(x)    (RADEG*acos(x))
#define atand(x)    (RADEG*atan(x))
```

```
#define atan2d(y,x) (RADEG*atan2((y),(x)))


double rev( double x )
{
    return  x - floor(x/360.0)*360.0;
}


double cbrt( double x )
{
    if ( x > 0.0 )
        return exp( log(x) / 3.0 );
    else if ( x < 0.0 )
        return -cbrt(-x);
    else /* x == 0.0 */
        return 0.0;
}
```

더 좋은 프로그래밍언어인 FORTRAN도 또한 표준라이브리와 함께 삼각함수들을 갖추어져 있습니다. 여기서 sin/cos/tan와 그의 역함수 또한 atan2도 지원하지만 단지 라디안 값에서만 지원됩니다. 그러므로 우리는 도값으로 삼각함수들을 얻을 수 있게 정의해야 합니다. 아래에 단지 SIND, ATAND, ATAN2D, REV와 CBRT만 주어져 있습니다. 나머지 COSD, TAND, ASINS와 ACOSD는 이와 유사한 방법으로 쓰여집니다.

```
      FUNCTION SIND(X)
      PARAMETER(RADEG=57.2957795130823)
      SIND = SIN( X * (1.0/RADEG) )
      END

      FUNCTION ATAND(X)
      PARAMETER(RADEG=57.2957795130823)
      ATAND = RADEG * ATAN(X)
      END

      FUNCTION ATAN2D(Y,X)
      PARAMETER(RADEG=57.2957795130823)
      ATAN2D = RADEG * ATAN2(Y,X)
      END

      FUNCTION REV(X)
      REV = X - AINT(X/360.0)*360.0
      IF (REV.LT.0.0)  REV = REV + 360.0
      END

      FUNCTION CBRT(X)
      IF (X.GE.0.0) THEN
        CBRT = X ** (1.0/3.0)
      ELSE
        CBRT = -((-X)**(1.0/3.0))
      ENDIF
      END
```

프로그래밍 언어 PASCAL은 삼각함수가 잘 갖추어져 있지 않습니다. PASCAL은 sin/cos/tan와 arctan만 있고 나머지 함수는 없습니다. 그러므로 arcsin, arccos, arctan를 만들 필요가 있고 도값에서의 삼각함수도 만들어야 합니다. 그리고 또한 rev()와 cbrt()함수도 만들어야 합니다. 도 값으로 정의되는 삼각함수는 다른 것들의 정의된 다음에 다시 정의됩니다. 그러므로 여기선 단지

arcsin, arccos, arctan2, rev와 cbrt만 정의하겠습다.

```
const pi      = 3.14159265358979323846;
      half_pi = 1.57079632679489661923;

function arcsin( x : real ) : real;
begin
  if x = 1.0 then
    arcsin := half_pi
  else if x = -1.0 then
    arcsin := -half_pi
  else
    arcsin := arctan( x / sqrt( 1.0 - x*x ) )
end;

function arccos( x : real ) : real;
begin
  arccos := half_pi - arcsin(x);
end;

function arctan2( y, x : real ) : real;
begin
  if x = 0.0 then
    begin
      if y = 0.0 then
        (* Error! Give error message and stop program *)
      else if y > 0.0 then
        arctan2 := half_pi
      else
        arctan2 := -half_pi
    end
  else
    begin
      if x > 0.0 then
          arctan2 := arctan( y / x )
      else if x < 0.0 then
        begin
          if y >= 0.0 then
            arctan2 := arctan( y / x ) + pi
          else
            arctan2 := arctan( y / x ) - pi
        end;
    end;
end;

function rev( x : real ) : real;
var rv : real;
begin
  rv := x - trunc(x/360.0)*360.0;
  if rv < 0.0 then
    rv := rv + 360.0;
  rev := rv;
end;

function cbrt( x : real ) : real;
begin
  if x > 0.0 then
    cbrt := exp ( ln(x) / 3.0 )
```

```
        else if x < 0.0 then
          cbrt := -cbrt(-x)
        else
          cbrt := 0.0
    end;
```

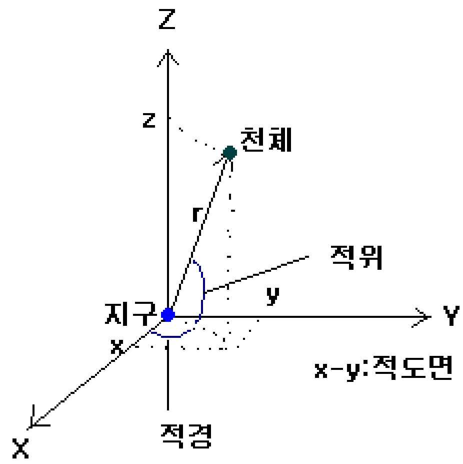
이런 이용할 수 있는 함수들을 만들어 내는 노력은 가치가 있습니다. 행성들의 위치를 계산하는 문제에서 실제로 사용되는 함수는 얼마 안되기 때문에 위에 정의한 함수말고 더 세부적인 것에 대해 생각할 필요는 없습니다.

다음으로는 직각좌표계와 구면좌표계에 관해서 논해보도록 하겠습니다. 이 좌표계들은 적도직각좌표계, 적도구면좌표계, 황도직각좌표계, 황도구면좌표계처럼 쓰여질 것입니다.

좌표계 coordinates

행성의 위치를 나타낼 수 있는 방법은 여러가지가 있는데 그중에서 여기서 사용 될 두가지 방법은 직각좌표계와 구면좌표계를 이용하는 것입니다.

여기선 특별히 적도좌표계에서의 직각좌표계와 구면좌표계를 논해보도록 하겠습니다. 한 행성이 어떤 RA, Decl, r값에 해당하는 위치에 있다고 가정합니다. 여기서 RA는 적경(Right Ascention), Decl은 적위(Declination), r은 지구중심에서 행성까지의 거리(distance)입니다. 만약 r이 알려져 있지 않거나 계산하기에 부적절값이라면 기본값 r=1로 놓아도 무방합니다. 직각좌표계의 좌표인 x,y,z값을 구해내기 위해 이들 값(RA,Decl,r)들을 이용할 것입니다.



직각좌표와 구면좌표관계

x = r * cos(RA) * cos(Decl)
y = r * sin(RA) * cos(Decl)
z = r * sin(Decl)

(RA의 cos나 sin값을 구해내기 위해 먼저 시/분/초로 이루어진 RA값을 시간과 그의 소수점이하값으로 변환시켜야 하며 그때 구해낸 시간값을 15를 곱하여 도값으로 변환시켜주어야 합니다.)

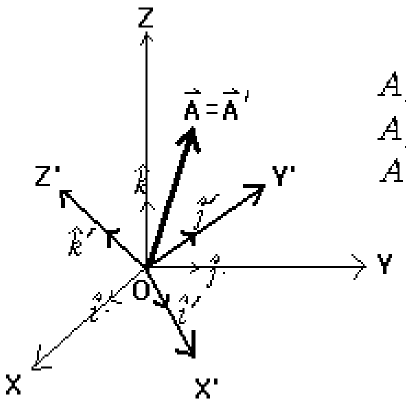
만약 위와는 반대로 직각좌표값(x,y,z)를 안다면 우리는 아래의 주어진 식을 이용하여 구면좌표계로 변환할 수 있습니다.

r = sqrt(x*x + y*y + z*z)
RA = atan2(y, x)
Decl = asin(z / r) = atan2(z, sqrt(x*x + y*y))

참고로 북극과 남극에서 천구의 극은 x,y값은 모두 0입니다. 그래서 atan2(0,0)을 계산할 수 없고 더불어 RA값도 구할 수 없습니다. 이것을 손쉽게 해결할 수 있는 방법은 RA를 어떤 임의의 값(예를 들어 0)으로 할당하는 것입니다.

구면좌표와 직각좌표로 변환하는 것은 적도좌표계에서 뿐만아니라, 황도좌표계와 지평좌표계에서도 적용될 수 있습니다. 즉, 각각 좌표계의 구면좌표값인 적경과 적위(적도좌표계), 황경와 황위(황도좌표계) 또는 방위각, 고도(지평좌표계)로 변환할 수 있다는 것입니다. 이것은 자명한 일입니다.

다음으로 좌표계의 회전에 대해서 논해보겠습니다. 좌표계를 회전시킨다는 것은 어떤 물체가 공간상에 정지해 있고 우리가 바라보는 눈의 방향을 X축이라고 할때 그 물체의 위치를 표시하는 것과 같습니다. 좌표계의 회전이 필요한 이유는 적도좌표에서 황도좌표 또는 지평좌표계로 변환시에 쓰여지기 때문입니다. 여기선 황도좌표계와 적도좌표계의 회전에 대해서 간단히 논해보도록 하겠습니다.



$$
\begin{aligned}
\vec{A} &= A_x\cdot\hat{i}+A_y\cdot\hat{j}+A_z\cdot\hat{k} \\
&= A_x\cdot\hat{i}\,'+A_y\cdot\hat{j}\,'+A_z\cdot\hat{k}\,' \\
A_x' &= \vec{A}\cdot\hat{i}\,'=(A_x\cdot\hat{i}+A_y\cdot\hat{j}+A_z\cdot\hat{k})\cdot\hat{i}\,' \\
A_y' &= \vec{A}\cdot\hat{j}\,'=(A_x\cdot\hat{i}+A_y\cdot\hat{j}+A_z\cdot\hat{k})\cdot\hat{j}\,' \\
A_k' &= \vec{A}\cdot\hat{k}\,'=(A_x\cdot\hat{i}+A_y\cdot\hat{j}+A_z\cdot\hat{k})\cdot\hat{k}\,'
\end{aligned}
$$

$$
\begin{pmatrix} A_x' \\ A_y' \\ A_k' \end{pmatrix} = \begin{pmatrix} i\cdot i' & j\cdot i' & k\cdot i' \\ i\cdot j' & j\cdot j' & k\cdot j' \\ i\cdot k' & j\cdot k' & k\cdot k' \end{pmatrix} \begin{pmatrix} A_x \\ A_y \\ A_k \end{pmatrix}
$$

좌표계의 회전

만약 직각좌표계가 X축을 기준으로 그 둘레를 회전하면 좌표계는 쉽게 새로운 X',Y',Z' 좌표계로 쉽게 계산할 수 있습니다. (일반적인 회전방법은 위의 그림에 주어져 있습니다.) 특별한 예로서 황도좌표 xeclip,yeclip,zeclip를 적도좌표 xequat,yequat,zequat로 회전한다고 가정합니다. 이 회전은 춘분점방향을 X축(황도좌표계와 적도좌표계에서는 흔히 춘분점방향을 원축으로 잡음)으로 하여 황도와 적도사이의 각도(the obliquity of the ecliptic, 대략 oblecl=23.4도)만큼 회전해서 새로운 좌표값을 구할 수 있는 것입니다.

xequat = xeclip
yequat = yeclip * cos(oblecl) - zeclip * sin(oblecl)
zequat = yeclip * sin(oblecl) + zeclip * cos(oblecl)

지금 구해진 xequat,yequat,zequat값은 적도좌표값이입니다. 이것은 쉽게 반대로 oblecl=23.5도만큼 회전해서 황도좌표값을 구할 수 있습니다.

xeclip = xequat
yeclip = yequat * cos(-oblecl) - zequat * sin(-oblecl)
zeclip = yequat * sin(-oblecl) + zequat * cos(-oblecl)

참고로 oblecl의 sin,cos을 계산할 때 다음과 같이 계산하면 됩니다.

cos(-x) = cos(x), sin(-x) = -sin(x)

그럼 실질적으로 위에서 배운 것을 토대로 구면황도좌표값(황경:long,황위:lat)로부터 구면적도좌표값(적경:RA,적위:Decl)으로 변환하는 예를 한번 들어보겠습니다. 이때 거리 r이 측정하기 어려운 값이라면 r=1로 놓습니다.

예 : 하지때 태양의 황도에서 황위는 90도입니다. 태양의 황위는 항상 거의 0에 가깝습니다. 먼저 이때 태양의 황도직각좌표값은 다음과 같습니다.

xeclip = cos(90도) * cos(0도) = 0.0000
yeclip = sin(90도) * cos(0도) = 1.0000
zeclip = sin(0도) = 0.0000

황도와 적도의 기울기가 23.4도라고 한다면 이 값을 황도좌표값이 적도좌표값이 되도록 x축을 기준으로 회전시켜보겠습니다.

xequat = 0.0000
yequat = 1.0000 * cos(23.4도) - 0.0000 * sin(23.4도)
zequat = 1.0000 * sin(23.4도) + 0.0000 * cos(23.4도)

여기서 적도직각좌표값들은 다음과 같이 됩니다.

xequat = 0
yequat = cos(23.4도) = 0.9178
zequat = sin(23.4도) = 0.3971

거리 r은 sqrt(0.8423 + 0.1577) = 1.0000이다 즉, 원래 거리와 변하지 않습니다. 이것은 자명한 일입니다.
다음으로 위에서 구한 적도직각좌표값을 적경, 적위값으로 다음과 같이 변환시켜보겠습니다.

RA = atan2(0.9178, 0) = 90도
Decl = asin(0.3971 / 1.0000) = 23.40도

양자택일적으로 아래와 같이 다른 함수를 써도 무방합니다.
Decl = atan2(0.3971, sqrt(0.8423 + 0.0000)) = 23.40도
여기서 우리는 적경 RA를 계산할 때 얼마나 간단하게 계산되는지 알 수 있었습니다. 여기선 되도록이면 asin()함수 대신해 atan2()함수를 쓰도록 하겠습니다. 그 이유는 atan2()함수는 이미 -180도에서 180도 범위에 나오도록 했음으로 결과값에 어느 사분면에 떨어질지 걱정하지 않아도 되기 때문입니다.

시간 척도. 시험 날짜 The time scale. A test date

여기서 사용되어지는 시간의 척도는 2000년 1월 0.0일 TDT로부터 '일수(day number)'입니다. 이것은 1999년 12월 31일 TDT(Terrestrial Dynamical Time)와 같습니다. 정확히 말하자면 이 세기의 마지막날의 시작의 자정 TDT라고 할 수 있습니다. 여기서 계산결과는 적당히 정확하면 됩니다(우리의 목표가 2분이내이므로). 그러므로 여기선 TDT와 UT(세계시)와의 차이점은 무시됩니다.

일수를 d라고 정의한다고 하면 이것은 율리우스 일(JD:Julian Day Number)또는 수정된 율리우스 일(MJD : Modified JD)을 사용하여 다음과 같이 계산할 수 있습니다.
d = JD - 2451543.5 = MJD - 51543.0
우리는 또한 달력으로부터 직접적으로 이것을 다음과 같이 계산할 수 있습니다.
d = 367*Y - (7*(Y + ((M+9)/12)))/4 + (275*M)/9 + D - 730530
Y는 년(4문자;예: 2000)이며, M은 달(1 - 12), 그리고 D는 날짜입니다. 이 식에서 모든 나눗셈은 정수 부분만을 취하면 됩니다. 파스칼에서는 나누는 부분인 "/"를 "div"로 사용하고 MS BASIC에서는 "₩"를 사용합니다. C/C++, FORTRAN에서는 "/"를 그대로 씁니다.

UT또는 TDT에 시작점인 자정에서 시작하여 정수인 d가 세어집니다. 만약 어떤 다른 시간을 원한다면 UT/24(여기서 나눗셈은 실수부분까지 취한다.)을 d에 더하여 사용할 수 있습니다.

지금까지의 일수 계산을 예를 들어봄으로써 한번 적용시켜보겠습니다.

예 : 1990년 4월 19일 0시 UT(세계시)라고 어떤 시험일자를 주겠습니다.

위에서 주어진 이 순간을 JD로 계산하면 JD=2448000.5에서 간단히 d=-3543.0를 얻을 수 있습니다.

또한 달력에 날짜로부터 직접적으로 다음과 같이 계산할 수 있습니다

d = 367*1990 - (7*(1990 + ((4+9)/12)))/4 + (275*4)/9 + 19 - 730530
d = 730330 - (7*(1990 + (13/12)))/4 + 1100/9 + 19 - 730530
d = 730330 - (7*(1990 + 1))/4 + 122 + 19 - 730530
d = 730330 - (7*1991)/4 + 122 + 19 - 730530
d = 730330 - 13937/4 + 122 + 19 - 730530
d = 730330 - 3484 + 122 + 19 - 730530 = -3543

이 1990년 4월 19일 0:00 UT의 순간은 다음부터 계속해서 쓸 것입니다. 1990년 4월 19일 우리가 정한 이 시험날짜의 d가 음수인 것은 1999년 12월 31일 원점보다 더 이전이라는 것을 알 수 있습니다.

궤도요소들 Orbital elements

앞에서 궤도요소들에는 주궤도요소와 부궤도요소들이 있다고 했습니다.
여기서는 주궤도요소에 대해서 다시한번 집어 넘어가고 각각 행성들에 대한 궤도요소값을 알아보도록 하겠습니다.

6가지 궤도요소를 정리하면 다음과 같습니다.

a : 평균거리 또는 장반경 - semi-major axis
e : 이심율 - eccentricity
T : 근일점 통과 시간
i : 기울기 - inclination

즉, 황도와 관련된 궤도의 기울기(tilt)를 말합니다. 기울기는 0에서 180도의 값을 가집니다. 만약 기울기가 90도보다 크면 행성은 역행을 하게 되는 것입니다. 즉 태양이 가는 반대방향으로 돌게 됩니다. 역행궤도를 가지는 잘 알려진 천체는 헬리혜성입니다.

N : (보통 "Omega:Ω" 문자로 쓰여집니다.) 승교점(천체가 황도면(기준면)을 기준으로 남반구에서 북반구로 지나갈때 천체의 궤도와 만나는 점. 즉, -위도에서 +위도로 넘어가는 찰라의 점)의 황경. 이 각은 춘분점으로부터 황도를 따라 승교점까지 각도를 말합니다.
w : (보통 "Small Omega:ω" 문자로 쓰여집니다.) 승교점으로부터 근일점까지의 천체의 궤도를 따라 잰 각.

아래에 주어지는 태양과 달, 행성들의 궤도요소들을 계산할 때 주의할 점이 있습니다. 만약 태양이나 달의 위치를 계산하면 다음 주어진 궤도요소는 지구를 중심으로해서 계산되어집니다. 그러나 다른 행성들의 위치는 태양 중심에 대해서 계산되어진다는 것입니다. 만약 이글을 읽고 계신 네티즌께서 각 행성들이 다른 행성들을 바라볼때 위치를 계산해야 하는데라고 걱정하실지도 모르겠습니다. 그러나 그건 별로 중요하지 않습니다. 나중에 궤도요소들을 통해서 위치가 계산되어지면 자동적으로 그런 계산은 간단히 할 수 있습니다. 그저 더하기 빼기만 알면 되지요. 이런 궤도요소들은 관측(천문측성학)에 의해 얻어진다는 사실도 알아두었으면 합니다.

그리고 또 한가지 알아두셔야 하는 것은 달의 거리는 지구의 반경으로 따진다는 점입니다. 태양이나 다른 행성들은 AU를 사용합니다.

아래의 궤도요소들을 보면 이상한 점이 있을 겁니다. 그것은 T(근일점 통과시간)이 없고 M(평균근점이각)이 있다는 점입니다. 이것은 각각의 궤도요소에 나와있는 d(일수, 전에 구했었죠?)로부터 구할 수 있기때문에 특별히 어떤 시간에 따른 행성의 위치를 정량적으로 표시해줄 수 있는 M이 주어진 것입니다. M은 나중에 이심근점이각, 진근점이각으로 계산되어 실질적인 행성의 위치를 계산하는데 밑거름이 되는 것입니다.

한가지 더 언급해야 하는 것은 M을 계산할때 자주 360도보다 크거나 음수인 수가 구해질 것입니다. 만약 음수인 경우에는 양수가 될때가지 360도를 더하고 360도보다 크면 360도 안에 들어올때까지 360도를 계속 빼주는게 좋습니다.

태양의 궤도요소들

    N = 0.0
    i = 0.0
    w = 282.9404 + 4.70935E-5 * d
    a = 1.000000 *(AU)*
    *e = 0.016709 - 1.151E-9 * d*
    *M = 356.0470 + 0.9856002585 * d*

달의 궤도요소들

N = 125.1228 - 0.0529538083 * d
i = 5.1454
w = 318.0634 + 0.1643573223 * d
a = 60.2666 *(지구반지름단위)*
e = 0.054900
*M = 115.3654 + 13.0649929509 * d*

수성의 궤도요소들

N = 48.3313 + 3.24587E-5 * d
i = 7.0047 + 5.00E-8 * d
w = 29.1241 + 1.01444E-5 * d
a = 0.387098 *(AU)*
*e = 0.205635 + 5.59E-10 * d*
*M = 168.6562 + 4.0923344368 * d*

금성의 궤도요소들

N = 76.6799 + 2.46590E-5 * d
i = 3.3946 + 2.75E-8 * d
w = 54.8910 + 1.38374E-5 * d
a = 0.723330 *(AU)*
*e = 0.006773 - 1.302E-9 * d*
*M = 48.0052 + 1.6021302244 * d*

화성의 궤도요소들

N = 49.5574 + 2.11081E-5 * d
i = 1.8497 - 1.78E-8 * d
w = 286.5016 + 2.92961E-5 * d
a = 1.523688 *(AU)*
*e = 0.093405 + 2.516E-9 * d*
*M = 18.6021 + 0.5240207766 * d*

목성의 궤도요소들

N = 100.4542 + 2.76854E-5 * d
i = 1.3030 - 1.557E-7 * d
w = 273.8777 + 1.64505E-5 * d
a = 5.20256 *(AU)*
*e = 0.048498 + 4.469E-9 * d*
*M = 19.8950 + 0.0830853001 * d*

토성의 궤도요소들

N = 113.6634 + 2.38980E-5 * d
i = 2.4886 - 1.081E-7 * d
w = 339.3939 + 2.97661E-5 * d
a = 9.55475 *(AU)*
*e = 0.055546 - 9.499E-9 * d*
*M = 316.9670 + 0.0334442282 * d*

천왕성의 궤도요소들

N = 74.0005 + 1.3978E-5 * d
i = 0.7733 + 1.9E-8 * d
w = 96.6612 + 3.0565E-5 * d
a = 19.18171 - 1.55E-8 * d *(AU)*
*e = 0.047318 + 7.45E-9 * d*
*M = 142.5905 + 0.011725806 * d*

해왕성의 궤도요소들

N = 131.7806 + 3.0173E-5 * d
i = 1.7700 - 2.55E-7 * d
w = 272.8461 - 6.027E-6 * d
a = 30.05826 + 3.313E-8 * d *(AU)*

*e = 0.008606 + 2.15E-9 * d*
*M = 260.2471 + 0.005995147 * d*

위에서 볼 수 있듯이 각각의 궤도요소에는 d(일수)가 있습니다. 전에 계산하는 방식으로 원하는 날짜의 d를 계산한 다음 이 궤도요소들의 대입만 해주면 그 날의 궤도요소가 나오게 됩니다.

해왕성과 천왕성간에 약 4200년 동안의 섭동주기가 있습니다. 그러므로 이들 요소들에 대해서 먼 과거나 먼 미래 위치의 정확성은 기대하기 어렵습니다.

처음에도 언급했다시피 명왕성의 위치계산은 수치적분법을 사용하게 됩니다. 그부분은 나중에 다루기로 하겠습니다.

태양의 위치 The Sun

이제부터 본격적으로 계산하도록 하겠습니다. 우리가 처음으로 계산할 것은 태양의 황경와 황위, 적경과 적위입니다.

오늘날 대부분 사람들은 지구가 태양주위를 돌고 있다는 것에 의심치 않습니다. 그러나 아래에 계산식은 마치 다른 방법으로 돌고 있는 것처럼 가장할 것입니다. 여기서 다른 방법은 태양이 지구를 돌고 있다는 가정한다는 것입니다. 이것은 지구를 중심으로 천체의 운동을 계산한다는 의도입니다. 그러나 나중에 지구중심좌표값을 태양중심좌표로 움직여 지구의 위치를 계산할 수도 있습니다. 단지 구해낸 지구중심직각좌표값에 모두 -1를 곱하면 그것이 태양중심직각좌표가 되는 것입니다. 여기서 표현되는 각도는 도값이고. 밑에는 태양의 위치를 계산하는데 필요한 태양궤도요소들이 있습니다.

w = 282.9404도 + 4.70935E-5도 * d (근일점의 경도)
a = 1.000000 (평균거리 단위 AU)
e = 0.016709 - 1.151E-9 * d (이심율)
M = 356.0470도 + 0.9856002585도 * d (평균근점이각)

우리는 또한 황도좌표계에서 적도좌표계로 변환시 황도의 기울기 Oblecl(the obliquity of the ecliptic)이 필요합니다. 그것은 다음과 같이 구해집니다.

oblecl = 23.4393도 - 3.563E-7도 * d

그리고 태양의 평균황경 L은 다음과 같처럼 주어질 수 있습니다.

L = w + M

정의에 의해 태양은 황도면을 움직이는 것처럼 보입니다. 그러므로 기울기 i는 0이고 그래서 승교점의 경도 N도 정의되어지지 않습니다. 여기선 N을 간단히 0으로 할당하도록 하겠습니다. 승교점과 근일점간사이의 각도 w는 근일점의 경도와 동등합니다.

지금부터 미리 정한 1990년 4월 19일 시험날짜에 태양의 위치를 계산하도록 하겠습니다. 전에 우리는 d=-3543.0을 계산했었습니다. 이 값을 이용해 계산하면 궤도요소들을 다음과 같은 값으로 계산할 수 있을겁니다.

w = 282.7735도
a = 1.000000
e = 0.016713
M = -3135.9347도

우리는 여기서 평균근점이각 M이 큰 음수값이라는 것을 주목할 필요가 있습니다. 이 값은 rev()함수를 이용하여 0과 360도사이로 만들어 놓습니다. 이 계산을 하기 위해서 rev()는 9*360=3240도를 구하여 M값에 더할 것입니다.

M = 104.0653도

우리는 또한

L = w + M = 386.8388도 = 26.8388도
oblecl = 23.4406도

이라는 것을 계산할 수 있습니다. 이제부터 보조각인 이심근점이각 E를 계산하도록 하겠습니다. 이것은 계속적인 급수형태로 계산되어 지는데 태양궤도의 이심율(지구궤도의 이심율)은 0.017으로 매우 작으므로 E의 첫번째 어림치는 충분히 정확하다고 할 수 있습니다(최소한 우리가 목표로 한 2분이내의 오차안에 넣을 수 있습니다). 참고로 달의 이심율은 0.05여서 좀 큰편입니다. 이럴 경우에는 반복풀이 방식이 있는데 그것은 다음에 설명하겠습니다.
아래 계산의 E와 M은 도값입니다.

E = M + (180/pi) * e * sin(M) * (1 + e * cos(M))

여기서 M과 e를 대입하면 E값을 계산할 수 있습니다.

E = 104.9904도

지금부터 근일점 방향쪽으로 X축으로한 황도좌표평면에 대한 직각좌표값을 계산하도록 하겠습니다.

x = r * cos(v) = cos(E) - e
y = r * sin(v) = sin(E) * sqrt(1 - e*e)

여기에 E를 대입하면

x = -0.275370
y = +0.965834

이것을 거리 r과 진근점이각 v로 변환하겠습니다.

r = sqrt(x*x + y*y)
v = arctan2(y, x)

수치적으로

r = 1.004323
v = 105.9134_deg

입니다. 이제 이것으로부터 태양의 황경를 계산할 수 있습니다.

lon = v + w
lon = 105.9134도 + 282.7735도 = 388.6869도 = 28.6869도

우리가 얻은 값이 실제값과 어느정도의 오차를 가질 것인가 비교해 보겠습니다.

	우리의 결과	실제관측값	오차
황경(lon)	28.6869도	28.6813도	+0.0056도 = 20"
거리(r)	1.004323 (AU)	1.004311(AU)	+0.000012(AU)

여기서 태양의 황경의 오차값은 20초입니다. 이 오차값은 우리가 목표로 한 1분이내 정확성을 가집니다. 거리를 따지면 지구지름의 약 1/3정도되므로 나쁘지 않은 계산결과입니다.

마지막으로 황도직각좌표계를 계산하고 이 값을 적도좌표계로 돌린뒤 태양의 적경(RA)과 적위(Decl)를 계산해보겠습니다.

x = r * cos(lon)
y = r * sin(lon)
z = 0.0

여기에 황경 lon을 대입하면

x = 0.881048
y = 0.482098
z = 0.0

전에 구했던 황도의 기울기 oblecl = 23.4406 도 만큼 황도좌표를 적도좌표로 돌리면

xequat = 0.881048
yequat = 0.482098 * cos(23.4406도) + 0.0 * sin(23.4406도)
zequat = 0.482098 * sin(23.4406도) + 0.0 * cos(23.4406도)

그러므로 결과는

xequat = 0.881048
yequat = 0.442312
zequat = 0.191778

이다. 이 값을 적경과 적위로 변환하면

r = 1.004323 (변하지 않았다.)
RA = 26.6580 도 = 26.6580/15 h = 1.77720 h = 1h 46m 37.9s
Decl = +11.0084 도 = +11_deg 0' 30"

실제관측값과 비교해 보겠습니다.

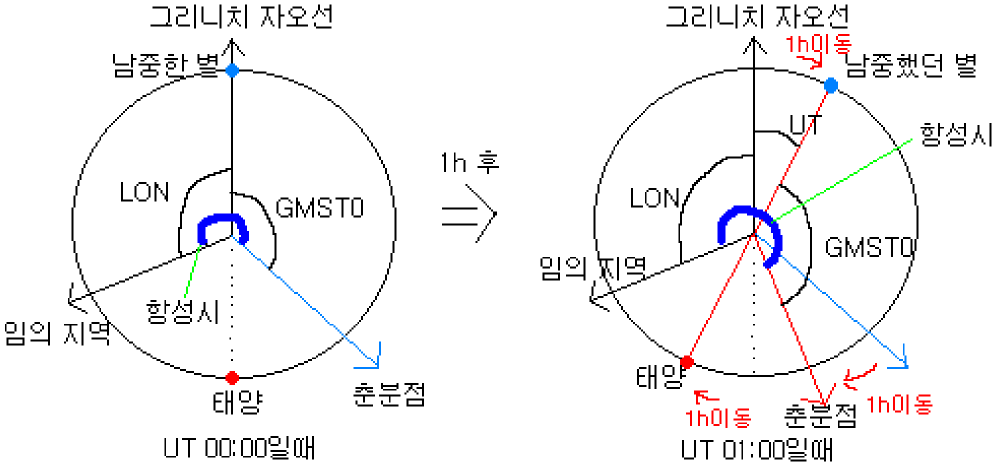
RA = 1h 46m 36.0s
Decl = +11_deg 0' 22"

어떻습니까? 차근차근 따라오면 그리 어려운 부분도 없을 겁니다. 다음으로 항성시와 시간각에 대해서 알아보고 이것을 이용하여 태양의 방위각과 고도를 계산해 보도록 하겠습니다.

항성시와 시간각. 고도와 방위각
Sidereal time and hour angle. Altitude and azimuth

항성시(Sidereal time)는 정확히 남쪽하늘부분의 자오선(meridian)의 적경을 말합니다. 즉, 항성시란 남중한 별의 적경과 같다고 할 수 있습니다. 항성시는 다음과 같이 계산할 수 있는 지방시(local time)입니다.(아래의 그림을 참고하기 바랍니다.)

SIDTIME = GMST0 + UT + LON/15



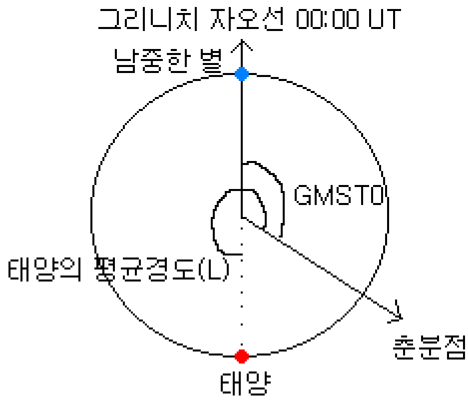
임의 지역의 항성시 구하기

여기에 SIDTIME, GMST0, UT값은 모두 시간단위로 주어지며 소수점도 생각합니다. GMST0는 영국의 그리니치 자오선상에서 00:00 자정에 항성시를 말합니다. UT는 그리니치 시간이며 LON은 지구상에서 관측자의 경도를 말합니다. 이 경도는 서쪽을 -, 동쪽을 +로 나타낸다. 이 경도값을 도값으로 표현하기 위해 15로 나누어 줍니다. 만약 항성시가 음수가 되면 24시간을 더하면 되고 24시를 초과하면 24시를 빼면됩니다.

어떻게 GMST0를 계산할 수 있을까? 간단합니다. 그것은 태양의 평균경도 L에 180도를 더하거나 빼면 됩니다. 그 결과를 0도에서 360도사이만 만들어주기 위해 rev()함수를 사용하고 마지막으로 15도 나누어 도값을 시간단위로 바꾸어줍니다.(식 아래 그림을 참고하기 바랍니다.)

GMST0 = (L + 180 도) / 15 = L/15 + 12h

여기서 이미 L=26.8388도 로 계산했으므로 대입하면,
GMST0 = 26.8388 도/15 + 12h = 13.78925 hours
이 됩니다. 지금부터 1990년 4월 19일 UT 00:00시의 유럽중심의 자오선(동경15도)상에서의 지방항성시를 계산하도록 하겠습니다.



GMST0(그리니치 항성시)

SIDTIME = GMST0 + UT + LON/15 = 13.78925h + 0 + 15도/15 = 14.78925 hours
SIDTIME = 14h 47m 21.3s

방위각과 고도를 구하기 위해 우리는 또 시간각(Hour Angle)을 알 필요가 있습니다. 시간각은 자오선상에서 0입니다. 즉 이때는 천체가 지평선에서 가장 높이 있을때 순간입니다.

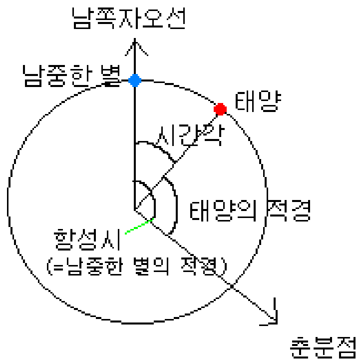
시간각은 시간이 지남에 따라 증가 합니다.(만약 물체가 지구자전보다 더 빠르게 움직이지 않는다면 이것은 대부분 인공위성의 경우에 해당된다.) 시간각은 다음과 같이 계산됩니다.(식 아래그림을 참고하기 바랍니다.)

HA = SIDTIME - RA

여기서 SIDTIME과 RA는 반드시 같은 단위(시간 또는 도값)으로 표시되어져야 합니다. 여기선 시간 단위를 선택하겠습니다.

HA = 14.78925h - 1.77720h = 13.01205h = 195.1808도

만약 시간각이 180도라면 천체는 북쪽(남반구에선 남쪽)에



항성시와 시간각

보일 것입니다(만약, 지평선 밑에 천체가 있다면 안보일 수도 있습니다). 여기서 우리는 지방시 01:00시쯤과 같은 크기인 HA = 195도를 얻었습니다.

지금 우리는 태양의 HA = 195.1808도 and Decl = +11.0084도를 가지고 천구의 적도의 남쪽을 X축으로 하고 지평선의 서쪽을 Y축, 천구의 북극을 Z축으로하는 직각좌표값 (x,y,z)를 구해보도록 하겠습니다. 여기서 거리 r이 불명확하다면 기본값 1로 놓도록 놓습니다.

x = cos(HA) * cos(Decl) = -0.947346
y = sin(HA) * cos(Decl) = -0.257047
z = sin(Decl) = +0.190953

지금부터 동서축 즉, Y축을 기준으로 위에서 구한 직각좌표계를 Z축이 천정에 위치하도록 돌릴것입니다. 여기서 북극에서 회전각도는 0입니다. 왜냐하면 북극은 이미 천정이 천구의 북극을 가리키고 있기 때문입니다. 북극말고 다른 위도를 가진 지방에선 90도-위도(latitudes)로 돌리게 됩니다.

xhor = x * cos(90도 - lat) - z * sin(90도 - lat)
yhor = y
zhor = x * sin(90도 - lat) + z * cos(90도 - lat)

sin(90도 - lat) = cos(lat)이므로 위의 식은 다음과 같습니다.

xhor = x * sin(lat) - z * cos(lat)
yhor = y
zhor = x * cos(lat) + z * sin(lat)

마지막으로 우리는 단순히 직각좌표에서 구면좌표로 바꿈으로써 방위각(azimuth)과 고도(altitude)를 계산합니다.

azimuth = atan2(yhor, xhor) + 180도
altitude = asin(zhor) = atan2(zhor, sqrt(xhor*xhor+yhor*yhor))

위에보면 방위각에 180도를 더한 것을 볼 수 있습니다. 이것은 지금 우리가 방위각을 북점(0도)에서 시계방향(동쪽,남쪽,서쪽순)으로 재기때문에 조절한 양입니다. 만약 이렇게 180도를 더하지 않는다면 남점이 0도가되어 서쪽, 북쪽, 동쪽순으로 재는것과 같아지게 될 것입니다.

우리는 스칸디나비아(Scandinavia) 중심부(동경 15도, 위도 북위 60도)를 선택했으므로 위의 식에 대입하면

xhor = -0.947346 * sin(60도) - (+0.190953) * cos(60도) = -0.915902
yhor = -0.257047 = -0.257047
zhor = -0.947346 * cos(60도) + (+0.190953) * sin(60도) = -0.308303

여기서 지평직각좌표계(the horizontal coordinates in rectangular form)를 계산했습니다. 이 값을 가지고 지평구면좌표계(the horizontal coordinates in spherical form), 즉 방위각과 고도를 계산하면 (거리 r=1)

azimuth = atan2(-0.257047,-0.915902) + 180도 = 375.6767도 = 15.6767도
altitude = asin(-0.308303) = -17.9570도

계산된 값을 소수점 3째짜리에서 반올림하면

azimuth = 15.68도, altitude = -17.96도

그러므로 스칸디나비아(Scandinavia) 중심부(동경 15도, 위도 북위 60도)에서 1990년 4월 19일 UT 00:00시에 태양은 지평선아래 17.96도에 있고 이 값은 천문박명(astronomical twilight)때인 지평선 18도 아래와 매우 가깝게 됩니다.

달의 위치 The Moon's position

달의 위치를 계산하도록 하겠습니다. 이번 계산은 달이 황도와 5도이상 더 기울어져 태양처럼 황도평면을 따라 회전하지 않기때문에 계산하는데 태양보다는 좀 더 복잡합니다. 또한 태양은 달의 운동에 영향을 주어 이때 생기는 섭동도 고려해야 합니다.

달의 궤도요소는 다음과 같습니다.
N = 125.1228도 - 0.0529538083도 * d (승교점의 황경)
i = 5.1454도 (기울기)
w = 318.0634도 + 0.1643573223도 * d (승교점과 근지점 사이 각도)
a = 60.2666 (지구 적도반경을 1로 할때 평균거리)
e = 0.054900 (이심율)
M = 115.3654도 + 13.0649929509도 * d (평균근점이각)

위에서 궤도요소를 보면 알 수 있겠지만 달의 승교점은 약 18.6년 주기로 역행방향으로 한번 공전하고 달의 근지점(지구와 가장 가까운 궤도의 지점)은 약 8.8년을 주기로 정방향으로 한번 공전하고 있습니다. 달 자신은 약 27.5년 주기로 지구를 한번 공전합니다. 평균거리 또는 장반축은 지구의 적도반경단위로 표현합니다.

최초에 선택한 시험날짜인 1990년 4월 19일(d=-3543)을 이용하여 달의 궤도요소를 수치적으로 계산해보겠습니다..
N = 312.7381도
i = 5.1454도
w = -264.2546도
a = 60.2666 (지구적도반경단위)
e = 0.054900
M = -46173.9046도

여기서 수치적 정확성이 얼마나 분명해야하는지 생각해볼려고 합니다. 만약 M이 단정도실수, 즉 단지 소수점 7자리까지만 계산한다면 M에따른 오차가 0.01도가 됩니다. 만약 1901년에서 2099년까지 중 어떤 날짜를 선택해서 계산하면 오차가 0.1도가 됩니다. 이것은 최초 목표로한 최고 오차 1내지 2분보다 더 크게 됩니다. 그러므로 달의 평균근점이각 M을 계산할 때는 적어도 9 또는 10자리까지의 정확성이 필요합니다.(마이크로컴퓨터가 드물었을때인 1980년에 위에서 설명한 문제가 발생했습니다. 그때는 포켓전자계산기가 마이크로컴퓨터보다 더 정확했습니다. 그러나 오늘날 마이크로컴퓨터는 배정도실수(14~16 소수점)를 제공하기에 계산하는데 문제가 없습니다.)

여기서 구한 모든 요소의 각도는 0~360도 안으로 만들어주기위해 rev() 함수를 써서 다음과 같이 계산합니다.
N = 312.7381도
i = 5.1454도
w = 95.7454도
a = 60.2666 (지구적도반경단위)
e = 0.054900
M = 266.0954도

여기서 M을 0~360도 안으로 만들어주기 위해 129*360 = 46440도를 더해야만 합니다.

다음으로 이심근점이각 E를 구해보도록 하겠습니다. 우리는 첫번째 근사치(E0)를 M을 이용해서 계산합니다.
E0 = M + (180도/pi) * e * sin(M) * (1 + e * cos(M))
달의 궤도의 이심율이 지구의 궤도의 이심율보다 더 큽니다. 이것은 위에서 구한 첫번째 근사치가 큰 오차를 낸다는 것을 의미합니다. 그러므로 더 정확한 계산을 하기위해 아래의 되풀이되는 식을 이용해 더욱더 정확한 값으로 수렴하도록 계산하고자 합니다. 최초 근사치 E0를 구하고 이 값을 아래식에 대입하면 새로운 E1이 나오게 하여 E0와 E1의 차이가 0.005도 이하가 되면 그때 구한 E1를 취하면 됩니다. 참고로 아래의 식은 도값을 사용하고 있습니다.
E1 = E0 - (E0 - (180도/pi) * e * sin(E0) - M) / (1 - e * cos(E0))
우리의 시험날짜에 E의 첫번째 근사치는 E=262.9689도입니다. 위의 식을 이용하여 되풀이하면 E = 262.9735도, 262.9735도...... 가 나옵니다.

다음엔 E를 이용하여 달까지 거리(r)와 진근점이각(v)를 구해보겠습니다. 먼저 달의 궤도평면상에 직각좌표값(x,y)를 구하겠습니다.

x = r * cos(v) = a * (cos(E) - e)
y = r * sin(v) = a * sqrt(1 - e*e) * sin(E)

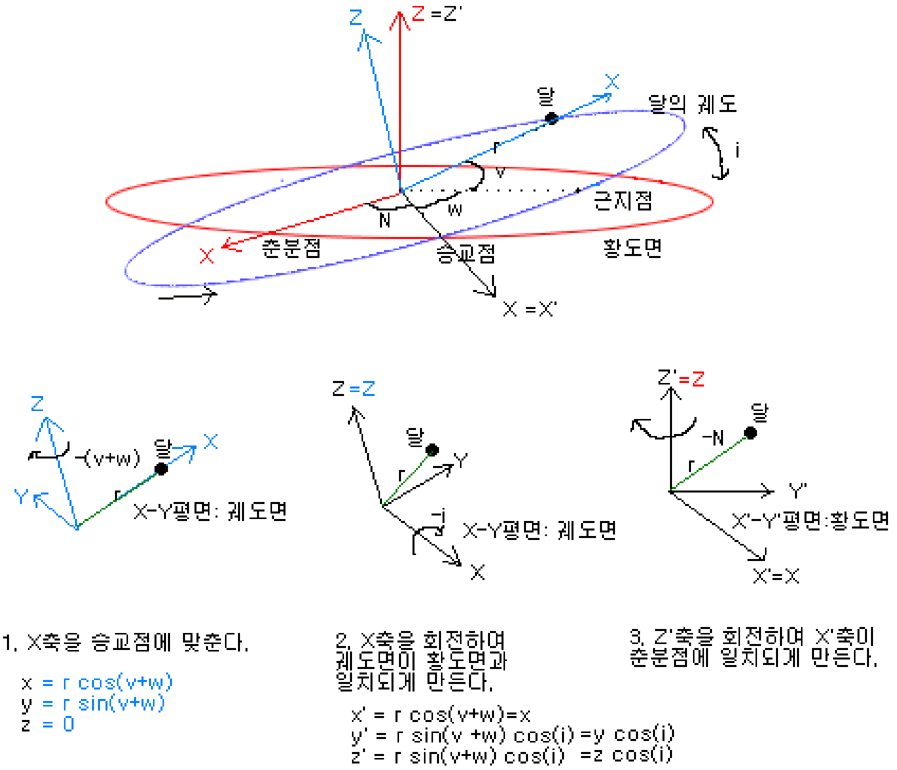
우리의 시험날짜에 위의 값은 다음과 같처럼 계산되어집니다.

x = -10.68095
y = -59.72377

다음으로 이것을 이용하여 달까지 거리(r)와 진근점이각(v)을 구하도록 하겠습니다.

r = sqrt(x*x + y*y) = 60.67134 지구반경단위
v = atan2(y, x) = 259.8605 도

달의 궤도평면에서 달의 위치를 알아내었습니다. 황도좌표계에서 달의 위치를 계산하기 위해 아래의 식을 적용하게 됩니다.(이것은 기본적으로 좌표계를 3번 돌리는 것과 같습니다. 첫번째 회전은 달의 궤도 좌표계의 X축, 즉 달의 방향 축을 승교점에 맞추기 위해 Z축을 돌립니다. 두번째 회전은 달의 궤도와 황도와의 기울기 i만큼 새로운 X축, 즉 승교점방향의 축을 돌립니다. 마지막으로 X축, 즉 승교점방향의 축이 춘분점과 일치시키기 위해 새로구해진 좌표계의 Z축을 승교점의 황경 N만큼 회전시킵니다. 이렇게 회전을 거듭한 좌표는 황도직각좌표값이됩니다. 아래그림을 참고하길 바랍니다. 다른 행성위치를 계산할 때도 이와같이 쓰므로 꼭 이해해야 합니다.)



달의 궤도 좌표에서 황도좌표로의 변환

xeclip = r * (cos(N) * cos(v+w) - sin(N) * sin(v+w) * cos(i))
yeclip = r * (sin(N) * cos(v+w) + cos(N) * sin(v+w) * cos(i))
zeclip = r * sin(v+w) * sin(i)

우리의 시험날짜에 위의 값은 다음과 같처럼 계산되어집니다.

xeclip = +37.65311
yeclip = −47.57180
zeclip = −0.41687

이 황도직각좌표값을 이용하여 황도좌표계에서 황경(longitude), 황위(latitude), 거리(distance)를 구합니다.

long = 308.3616도
lat = −0.3937도
r = 60.6713

실제 관측값을 보면 달의위치는 이 순간 황경이 306.94도. 황위는 −0.55도입니다. 이것은 우리가 계산한 값과 각각 1.42도, 0.16도나 차이가 났습니다! 이 차이는 우리가 겨냥한 목표인 1~2분이내 계산을 실패한 것처럼보입니다. 과연 그럴까요???

섭동을 고려한 더 정확한 달의 위치 The Moon's position

우리가 계산한 달의 위치 계산에 큰 오차가 생기는 이유는 달에 작용되는 섭동을 무시했기 때문입니다. 우리는 다뤄야하는 대부분 중요한 섭동항을 아래에서 계산할 것입니다. 여기에서 계산한 결과를 전에 계산한 것에 더하여 우리가 목표로한 오차범위 안에 줄이고자 합니다.

먼저 우리는 몇가지 기초적인 변수들이 필요합니다.

태양의 평균 황경: Ls (이미 계산되었다)
달의 평균 황경: Lm = N + w + M (for the Moon)
태양의 평균근점이각: Ms (이미 계산되었다)
달의 평균근점이각: Mm (이미 계산되었다)
달의 평균이각: D = Lm - Ls
달의 위도의 변수 : F = Lm - N

우리의 시험날짜에 이 식들을 계산해보겠습니다.

Ms = 104.0653도
Mm = 266.0954도
Ls = 26.8388도
Lm = 312.7381도 + 95.7454도 + 266.0954도 = 674.5789도 = 314.5789도
D = 314.5789도 - 26.8388도 = 287.7401도
F = 314.5789도 - 312.7381도 = 1.8408도

지금부터 황경에서 가장 큰 섭동항 12개, 황위에서 가장 큰 섭동항 5개, 거리에서 2개를 더하여 계산할 것입니다. 여기서 소개되는 섭동항들은 모두 황경과 황위를 0.01도보다 더 크게 하는 것입니다. 그리고 거리에서 섭동항은 지구의 반경의 0.1배보다 더 크게하는 항입니다.

황경의 섭동(도값):
−1.274_deg * sin(Mm − 2*D) (출차(出差), Evection)
+0.658_deg * sin(2*D) (변차(變差), Variation)
−0.186_deg * sin(Ms) (연차(年差), Yearly equation)
−0.059_deg * sin(2*Mm − 2*D)
−0.057_deg * sin(Mm − 2*D + Ms)
+0.053_deg * sin(Mm + 2*D)
+0.046_deg * sin(2*D − Ms)
+0.041_deg * sin(Mm − Ms)
−0.035_deg * sin(D) (월각차(月角差), Parallactic equation)
−0.031_deg * sin(Mm + Ms)
−0.015_deg * sin(2*F − 2*D)

+0.011_deg * sin(Mm - 4*D)

황위의 섭동(도값):
-0.173_deg * sin(F - 2*D)
-0.055_deg * sin(Mm - F - 2*D)
-0.046_deg * sin(Mm + F - 2*D)
+0.033_deg * sin(F + 2*D)
+0.017_deg * sin(2*Mm + F)

거리에 대한 섭동(지구반경):
-0.58 * cos(Mm - 2*D)
-0.46 * cos(2*D)

황도에 대한 가장 큰 섭동항에 몇몇은 그 항의 개개의 이름이 주어졌습니다.
가장 큰 섭동항인 태양 인력에 의한 달 운행의 주기적 차이인 출차(出差, Evection)는 이미 Ptolemy에 의해 발견되었습니다. 그는 이 항을 그의 이론인 천동설에 주전원의 하나로 만들었습니다.

그 다음 큰 섭동항인 변차(變差, Variation)와 연차(年差, Yearly equation)는 티코 브라헤에 의해 발견되었습니다. 여기서 연차는 지구가 근일점(近日點)을 통과하여 다시 근일점으로 돌아오기까지의 시간인 1근점년(近點年, anomalistic year)이 지구의 근일점이 매년 11.63"씩 황도를 따라 진행하게되어 근점년은 항성년보다 약 0.00328일이 길어, 365.25964일이 되는데 이러한 차이로 인해 달의 황경에 변화를 주는 것을 말합니다. 또 변차도 또한 달의 황경운동에 대한 부등(不等)의 하나로 진폭이 0.66도이고 주기는 반삭망월(14.7653일)인 섭동항입니다.
또 위에 월각차(月角差, Parallactic equation)라는 것이 있는데 이것도 또한 달의 황경을 변화시키는 것으로 −125" sin(Lm−Ls)의 형태로 표시됩니다. 아시다시피 Lm은 달의 평균황경, Ls는 태양의 평균황경이며, Lm=Ls일 때가 삭(朔)이기 때문에 월각차의 주기는 1삭망월(朔望月)이 됩니다. 월각차로 인해 달의 위치는 삭에서 망까지는 평균위치보다 늦어지며, 망에서 삭까지는 앞당겨집니다.

월각차가 잘 알려진 것은 계수 125"(=-0.035도)가 태양의 지평시차(地平視差)의 정확하게 14.214배가 되어야 한다는 것이 이론상 증명되었기 때문입니다. 1948년 D.브라우어가 성식관측(星蝕觀測)을 정리하여 결정한 계수는 124.997"로, 이것에 의하면 태양의 시차(視差)는 8.794"가 됩니다.

만약 1~2분이내에 정확성이 필요없다면 위의 섭동항을 계산하지 않아도 됩니다. 또 황경은 0.25도 내외 황위를 0.15 내외의 오차를 허용한다면 위에 주어진 가장큰 항 2개만 포함하면 됩니다

자, 이제부터 우리의 시험일자로 이 섭동항들을 계산해 보겠습니다.

    longitude(황경): -0.9847 - 0.3819 - 0.1804 + 0.0405 - 0.0244 + 0.0452 +
                     0.0428 + 0.0126 - 0.0333 - 0.0055 - 0.0079 - 0.0029
                     = -1.4132도

    latitude(황위): -0.0958 - 0.0414 - 0.0365 - 0.0200 + 0.0018 = -0.1919도

    distance(거리): -0.3680 + 0.3745 = +0.0066 지구반경단위

이미 계산된 달의 황도좌표값에 계산한 값을 더하면 아래와 같습니다.

    long = 308.3616도 - 1.4132도 = 306.9484도
    lat = -0.3937도 - 0.1919도 =   -0.5856도
    dist = 60.6713 + 0.0066   =   60.6779 지구반경단위

실제값과 비교해봅니다.

    longitude 306.94도, latitude -0.55도, distance 60.793 지구반경단위

더욱 나아졌죠?

이제 한번 이들 황도좌표값으로부터 적도좌표값인 적경과 적위로 변환시켜보겠습니다. 우리는 일찍이 황도좌표값 황경과 황위를 직각좌표값 x,y,z 을 구하여 이것을 황도의 기울기 만큼 이 좌표계를 돌렸습니다. 그리하여 구해진 값을 구면좌표값으로 변환하여 적도좌표값인 적경과 적위, 그리고 거리를 구했습니다. 여기서 달의 거리는 황도좌표값과 적도좌표값 똑같을 것입니다. 다시말해 r=1 로써 일정합니다.

RA   = 309.5011도
Decl = -19.1032도

실제값과 비교해봅니다.

RA = 309.4881도
Decl = -19.0741도

어떻습니까? 이제 섭동항을 계산함으로써 더 정확한 달의 위치를 계산할 수 있었습니다.

지표면에서의 달의 위치 The Moon's topocentric position

지금가지 계산한 달의 위치는 지구중심(geocentric)에서의 위치입니다. 즉 관측자가 지구 중심에 있다고 가정했을시의 위치라고 할 수 있습니다. 그러나 실제 관측자는 지구의 표면에 거주하고 있기 때문에 지구중심에서의 달의 위치와 지표면에서의 달의 위치가 달라질 것입니다. 여기선 지표면에서의 달의 위치를 계산해보도록 하겠습니다.

지구중심에서 달의 위치는 일찍이 계산되어졌습니다(Geocentric Equatorial coordinates). 그러나 관측자는 지구중심이 아닌 지표면 위에 있습니다. 그러므로 달의 위치가 바뀔 수 밖에 없습니다.

그러므로 이런 관계에 관해 생각해 볼 필요가 있스비다. 참고로 우리가 목표로한 1~2붙이내의 위치계산을 위해선 다른 행성들까지 이런 관계를 주어질 필요가 없습니다.

먼저 이 계산을 하기 위해 알아야 될 것을 아래에 설명하겠습니다.

r: 지구중심에서 달까지 거리(지구반경단위)
rho : 지구반경(기본값:1)
lat : 관측자의 관측적 위도(즉, 북극성의 고도를 말한다.)
gclat : 관측자의 실제 위도(이것은 지구 중심에서의 관측자의 실제 위로들 말한다.)
LST : 관측자가 있는 지방의 지방항성시(Local Sidereal time)
gecRA : 지구중심에서 달의 적경
gecDecl : 지구중심에서의 달의 적위

rho와 lat, gclat는 지구가 완전구라면 rho=0, lat=gclat라고 할 수 있다. 그러나 지구는 구가 아닌 적도쪽으로 찌그러진 타원체(편평도 약 1/300)이므로 이것을 감안하여 rho와 gclat를 계산해야 된다.

계산은 다음과 같이 한다.

gclat = lat - 0.19214도 * sin(2*lat)
rho = 0.99883 + 0.00167 * cos(2*lat)

위와같이 관측자의 rho와 gclat를 계산하고 미리구해진 시험날짜에 관측자의 LST를 가지고 지표면의 위치를 구할 수 있다. 즉, 지표면 좌표계의 원점(O')의 위치를 지구중심에서 바라볼때 위치로 구할 수 있다.

이것은 회전이아닌 그저 좌표계의 평행이동으로 한다. 이것은 이 구해진 함수중 직각좌표계로변환하는 함수(ConvertToRectangular(거리, 경도, 위도)에 대입하면 지표면의 우치가 구해진다. 그리고 기존에 구한 지구중심에서 달의 적도좌표값을 빼면 바로 지표면에서의 적도좌표값(Topocentric equatorial coodinates)를 구할 수 있다.

topoEquatY = geoEquX - ObspositionX
topoEquatY = geoEquY - ObspositionY
topoEquatZ = geoEquZ - ObspositionZ

이 값을 ConverToSperical(x,y,z)함수를 이용해 달의 r, gecRA, gecDecl를 구하면 된다.

행성의 궤도요소 계산
Computing the orbital elements of the planets

지금부터 행성의 궤도요소를 계산해보도록 하겠습니다. 행성의 궤도요소도 달과 태양의 궤도요소를 계산한 것처럼 우리가 정한 시험날짜인 1990년 4월 19일 00:00 UT를 기준으로 계산하면 다음과 같습니다.

궤도요소 / 행성	N	i	w	a	e	M
	도	도	도	AU		도
수성(Mercury)	48.2163	7.0045	29.0882	0.387098	0.205633	69.5153
금성(Venus)	76.5925	3.3945	54.8420	0.723330	0.006778	131.6578
화성(Mars)	49.4826	1.8498	286.3978	1.523688	0.093396	321.9965
목성(Jupiter)	100.3561	1.3036	273.8194	5.20256	0.048482	85.5238
토성(Saturn)	113.5787	2.4890	339.2884	9.55475	0.055580	198.4741
천왕성(Uranus)	73.9510	0.7732	96.5529	19.18176	0.047292	101.0460
해왕성(Neptune)	131.6737	1.7709	272.8675	30.05814	0.008598	239.0063

행성들의 태양중심위치 The heliocentric positions of the planets

행성들의 태양중심에서의 위치도 또한 달과 태양의 위치를 계산한 지구중심에서의 위치계산할 때와 같습니다. 먼저 이심근점이각 E를 계산합니다. 이때 이심율이 크면 달의 이심근점이각을 계산한 것과 같이 되풀이되는 방식으로 계산해 줍니다. E를 알게되면 E로 부터 진근점이각 v를 구합니다. 이 값을 가지고 황도직각좌표값 x,y,z값을 구합니다. 이때 좌표값들은 태양과 달의 위치를 계산한 것과 달리 태양중심에서의 좌표값을 가리킵니다. 또한 달의 장반경 a, 거리 r은 모두 지구반경단위였지만 여기서의 이들 값은 천문단위 AU입니다. 즉, 1AU는 약 146.6백만 km인 셈입니다.

그럼 우리가 정한 시험날짜로부터 수성의 이들 값들을 계산해 보겠습니다.

먼저 수성의 이심근점이각의 어림치 E는 81.3464도 입니다. 계속적인 되풀이 계산을 하면 81.1572도, 81.1572도 . . . . 가 되어 이것으로부터 다음과 같은 값을 계산할 수 있습니다.

r = 0.374862 AU
v = 93.0727도

수성의 태양중심황도직각좌표값(heliocentric ecliptic rectangular coordinates)은 다음과 같이 됩니다.

x = -0.367821 AU
y = +0.061084 AU
z = +0.038699 AU

구면좌표값으로 변환하면 다음과 같습니다.

lon = 170.5709 도
lat = +5.9255 도
r   = 0.374862 AU

실제값과 비교해 봅니다

lon = 170.5701도
lat = +5.9258도
r = 0.374856 AU

이 값은 거의 완벽합니다. 오차간 단 몇초에 불과합니다. 이것은 수성이 태양에 아주 가까이 있어서 다른 행성들에 의한 중력이 태양의 중력에 비해 아주 작으므로 태양의 중력의 영향이 훨씬 커서 섭동을 일으키지 않는 것입니다. 그래서 수성의 위치를 계산하는 것은 다른 행성들보다 더 쉽습니다.

만약 태양중심에서의 황경과 황위, 거리를 궤도요소들로부터 구한다면 다음과 같은 값을 얻을 것입니다.

태양중심에서의 황경과 황위, 거리	태양중심		
	황경(lon)	황위(lat)	거리(r)
수성(Mercury)	170.5709 도	+5.9255 도	0.374862 AU
금성(Venus)	263.6570 도	-0.4180 도	0.726607 AU
화성(Mars)	290.6297 도	-1.6203 도	1.417194 AU
목성(Jupiter)	105.2543 도	+0.1113 도	5.19508 AU
토성(Saturn)	289.4523 도	+0.1792 도	10.06118 AU
천왕성(Uranus)	276.7999 도	-0.3003 도	19.39628 AU
해왕성(Neptune)	282.7192 도	+0.8575 도	30.19284 AU

토성의 실제값과 비교해 봅니다.

lon = 289.3864 도
lat = +0.1816 도
r = 10.01850 AU

여기선 너무 많은 차이가 있다는 것을 알수 있습니다. 수성은 오차가 몇초밖에 안되나 토성은 그 오차가 4분보다 더 큽니다. 이 오차는 가끔 1도 이상의 오차도 내기도 합니다. 이것은 거의 목성에 의한 섭동영향입니다.

섭동을 고려한 더 정확한 행성들의 위치
Higher accuracy - perturbations

우리가 목표로한 1~2분이내의 정확성을 가지기 위해 우리는 목성, 토성, 천왕성의 각각 상호간에 섭동을 생각해야 합니다. 만약 1~2분이내의 정확성을 원치않는다면 지금부터 계산된 섭동항들을 무시해도 됩니다.

먼저 기초값이 필요합니다.

목성의 평균근점이각: Mj
토성의 평균근점이각: Ms
천왕성의 평균근점이각: Mu

다음 항을 목성의 태양중심 황경에 더합니다.

```
-0.332_deg * sin(2*Mj - 5*Ms - 67.6_deg)
-0.056_deg * sin(2*Mj - 2*Ms + 21_deg)
+0.042_deg * sin(3*Mj - 5*Ms + 21_deg)
-0.036_deg * sin(Mj - 2*Ms)
+0.022_deg * cos(Mj - Ms)
+0.023_deg * sin(2*Mj - 3*Ms + 52_deg)
```

-0.016_deg * sin(Mj - 5*Ms - 69_deg)

다음 항들을 토성의 황경과 황위에 더합니다.

토성의 태양중심 황경에 대한 섭동항들...

+0.812_deg * sin(2*Mj - 5*Ms - 67.6_deg)
-0.229_deg * cos(2*Mj - 4*Ms - 2_deg)
+0.119_deg * sin(Mj - 2*Ms - 3_deg)
+0.046_deg * sin(2*Mj - 6*Ms - 69_deg)
+0.014_deg * sin(Mj - 3*Ms + 32_deg)

토성의 태양중심 황위에 대한 섭동항들...

-0.020_deg * cos(2*Mj - 4*Ms - 2_deg)
+0.018_deg * sin(2*Mj - 6*Ms - 49_deg)

마지막으로 다음 나오는 섭동항들을 천왕성의 태양중심 황경값에 더합니다.

+0.040_deg * sin(Ms - 2*Mu + 6_deg)
+0.035_deg * sin(Ms - 3*Mu + 33_deg)
-0.015_deg * sin(Mj - Mu + 20_deg)

이 섭동항들은 모두 0.01도 이상의 진폭을 가지게 합니다. 여기선 행성들의 거리에 대한 섭동은 무시합니다. 이들 섭동은 겉보기위치에 별로 영향을 받지 않습니다.

가장 큰 섭동항인 "목성-토성 항"은 황경에 가장 큰 진폭의 섭동을 만들어 냅니다. 이 주기는 918년이고 목성, 토성간에 진폭은 몇도이상 차이가 납니다. 다음으로는 "토성-천왕성 항"인데 이것은 주기가 560년이고 그 진폭은 천왕성에 대해서 0.035도이나 토성은 0.01도보다 더 작습니다. 다른 항들은 14년에서 100년 사이의 주기를 가집니다. 마지막으로 "천왕성-해왕성 항"인데 이것은 주기가 4200년이고 거의 1도정도의 진폭을 가집니다. 이 섭동항은 여기서 다루지 않고 대신해 이미 천왕성과 해왕성의 궤도요소안에 포함시켰습니다. 이것은 해왕성과 천왕성의 평균거리가 변하기 때문입니다.

이제 우리의 시험날짜에 이들 섭동을 계산해 보도록 하겠습니다. 먼저 기초값으로

Mj = 85.5238_deg   Ms = 198.4741_deg   Mu = 101.0460:

목성의 황경에 대한 섭동항값은

+ 0.0637_deg - 0.0236_deg + 0.0038_deg - 0.0270_deg - 0.0086_deg
- 0.0049_deg - 0.0155_deg = -0.0120_deg

목성의 섭동을 고려한 태양중심 황경값 : 105.2423도
실제값 : 105.2603도

토성의 황경에 대한 섭동항값은

-0.1560_deg + 0.0206_deg + 0.0850_deg - 0.0070_deg - 0.0124_deg
= - 0.0699_deg

토성의 황위에 대한 섭동항값은

+0.0018_deg + 0.0035_deg = +0.0053_deg

토성의 섭동을 고려한 태양중심 황경, 황위값 : 289.3824도, +0.1845도
실제값 : 289.3864도, +0.1816도

천왕성의 황경에 대한 섭동항값은

    +0.0017_deg - 0.0332_deg - 0.0012_deg = -0.0327_deg

천왕성의 섭동을 고려한 태양중심 황경, 황위값 : 276.7672도
실제값 : 276.7706도

Precession

The planetary positions computed here are for "the epoch of the day", i.e. relative to the celestial equator and ecliptic at the moment. Sometimes you need to use some other epoch, e.g. some standard epoch like 1950.0 or 2000.0. Due to our modest accuracy requirement of 1-2 arc minutes, we need not distinguish J2000.0 from B2000.0, it's enough to simply use 2000.0.

We will simplify the precession correction further by doing it in eliptic coordinates: the correction is simply done by adding

    3.82394E-5_deg * (365.2422 * (epoch - 2000.0) - d)

to the ecliptic longitude. We ignore precession in ecliptic latitude.
"epoch" is the epoch we wish to precess to, and "d" is the "day number" we used when computing our planetary positions.

Example: if we wish to precess computations done at our test date 19 April 1990, when d = -3543, we add the quantity below (degrees) to the ecliptic longitude:

    3.82394E-5_deg * (365.2422 * (2000.0 - 2000.0) - (-3543)) =
    = 0.1355_deg

So we simply add 0.1355_deg to our ecliptic longitude to get the position at 2000.0.

Geocentric positions of the planets

To convert the planets' heliocentric positions to geocentric positions, we simply add the Sun's rectangular (x,y,z) coordinates to the rectangular (x,y,z) heliocentric coordinates of the planet:

Let's do this for Mercury on our test date - we add the x, y and z coordinates separately:

```
    xsun  = +0.881048    ysun  = +0.482098    zsun  = 0.0
    xplan = -0.367821    yplan = +0.061084    zplan = +0.038699
-----------------------------------------------------------------
    xgeoc = +0.513227    ygeoc = +0.543182    zgeoc = +0.038699
```

Now we have rectangular geocentric coordinates of Mercury. If we wish, we can convert this to spherical coordinates - then we get geocentric ecliptic longitude and latitude. This is useful if we want to precess the position to some other epoch: we then simply add the approproate precessional value to the longitude. Then we can convert back to rectangular coordinates.

But for the moment we want the "epoch of the day": let's rotate the x,y,z, coordinates around the X axis, as described earlier. Then we'll get equatorial rectangular geocentric (whew!) coordinates:

    xequat = +0.513227   yequat = +0.482961   zequat = 0.251582

The elongation and physical ephemerides of the planets

Let's start by computing the apparent diameter of the planet:

$$d = d0 / R$$

R is the planet's geocentric distance in astronomical units, and d is the planet's apparent diameter at a distance of 1 astronomical unit. d0 is of course different for each planet. The values below are given in seconds of arc. Some planets have different equatorial and polar diameters:

Mercury	6.74"			
Venus	16.92"			
Earth	17.59"	equ	17.53"	pol
Mars	9.36"	equ	9.28"	pol
Jupiter	196.94"	equ	185.08"	pol
Saturn	165.6"	equ	150.8"	pol
Uranus	65.8"	equ	62.1"	pol
Neptune	62.2"	equ	60.9"	pol

The Sun's apparent diameter at 1 astronomical unit is 1919.26". The Moon's apparent diameter is:

$$d = 1873.7'' * 60 / r$$

where r is the Moon's distance in Earth radii.

Two other quantities we'd like to know are the phase angle and the elongation.

The phase angle tells us the phase: if it's zero the planet appears "full", if it's 90 degrees it appears "half", and if it's 180 degrees it appears "new". Only the Moon and the inferior planets (i.e. Mercury and Venus) can have phase angles exceeding about 50 degrees.

The elongation is the apparent angular distance of the planet from the Sun. If the elongation is smaller than ca 20 degrees, the planet is hard to observe, and if it's smaller than ca 10 degrees it's usually not possible to observe the planet.

To compute phase angle and elongation we need to know the planet's heliocentric distance, r, its geocentric distance, R, and the distance to the Sun, s. Now we can compute the phase angle, FV, and the elongation, elong:

$$elong = acos((s*s + R*R - r*r) / (2*s*R))$$

$$FV = acos((r*r + R*R - s*s) / (2*r*R))$$

When we know the phase angle, we can easily compute the phase:

$$phase = (1 + cos(FV)) / 2 = hav(180\_deg - FV)$$

hav(FV) is the "haversine" of the phase angle. The "haversine" (or "half versine") is an old and now obsolete trigonometric function; it's defined as:

$$hav(x) = (1 - cos(x)) / 2 = sin^2 (x/2)$$

As usual we must use a different procedure for the Moon. Since the Moon is so close to the Earth, the procedure above would introduce too big errors. Instead we use the Moon's ecliptic longitude and latitude, mlon and mlat, and the Sun's ecliptic longitude, mlon, to compute first the elongation, then the phase angle, of the Moon:

```
elong = acos( cos(slon - mlon) * cos(mlat) )

FV = 180_deg - elong
```

Finally we'll compute the magnitude (or brightness) of the planets. Here we need to use a formula that's different for each planet. The phase angle, FV, is in degrees:

```
Mercury:   -0.36 + 5*log10(r*R) + 0.027 * FV + 2.2E-13 * FV**6
Venus:     -4.34 + 5*log10(r*R) + 0.013 * FV + 4.2E-7  * FV**3
Mars:      -1.51 + 5*log10(r*R) + 0.016 * FV
Jupiter:   -9.25 + 5*log10(r*R) + 0.014 * FV
Saturn:    -9.0  + 5*log10(r*R) + 0.044 * FV + ring_magn
Uranus:    -7.15 + 5*log10(r*R) + 0.001 * FV
Neptune:   -6.90 + 5*log10(r*R) + 0.001 * FV
```

** is the power operator, thus FV**6 is the phase angle (in degrees) raised to the sixth power. If FV is 150 degrees, then FV**6 becomes ca 1.14E+13, which is a quite large number.

Saturn needs special treatment due to its rings: when Saturn's rings are "open" then Saturn will appear much brighter than when we view Saturn's rings edgewise. We'll compute ring_mang like this:

```
ring_magn = -2.6 * sin(abs(B)) + 1.2 * (sin(B))**2
```

Here B is the tilt of Saturn's rings which we also need to compute. Then we start with Saturn's geocentric ecliptic longitude and latitude (los, las) which we've already computed. We also need the tilt of the rings to the ecliptic, ir, and the "ascending node" of the plane of the rings, Nr:

```
ir = 28.06_deg
Nr = 169.51_deg + 3.82E-5_deg * d
```

Here d is our "day number" which we've used so many times before. For our test date d = -3543. Now let's compute the tilt of the rings:

```
B = asin( sin(las) * cos(ir) - cos(las) * sin(ir) * sin(los-Nr) )
```

This concludes our computation of the magnitudes of the planets.

The positions of comets. Comet Encke and Levy.

If you want to compute the position of a comet or an asteroid, you must have access to orbital elements that still are valid. One set of orbital elements isn't valid forever. For instance if you try to use the 1986 orbital elements of comet Halley to compute its appearance in either 1910 or 2061, you'll get very large errors in your computed positions - sometimes the errors will be 90 degrees or more.

Comets will usually have a new set of orbital elements computed for each perihelion. The comets are perturbed most severely when they're close to aphelion, far away from the gravity of the Sun but maybe much closer to Jupiter, Saturn, Uranus or Neptune. When the comet is passing through the inner solar system, the perturbations are usually so small that the same set of orbital elements can be used for the entire apparition.

Orbital elements for an asteroid should preferably not be more than about one year old. If your accuracy requirements are lower, you can of course use older elements. If you use orbital elements that are five years old for a main-belt asteroid, then your computed positions can be several degrees in error. If the orbital elements are less than one year old, the errors usually stay below approximately one arc minute, for a main-belt asteroid.

If you have access to valid orbital elements for a comet or an asteroid, proceed as below to compute its position at some date:

1. If necessary, precess the angular elements N,w,i to the epoch of today. The simples way to do this is to add the precession angle to N, the longitude of the ascending node. This method is approximate, but it's good enough for our accuracy aim of 1-2 arc minutes.

2. Compute the day number for the time or perihelion, call it D. Then compute the number of days since perihelion, d - D (before perihelion this number is of course negative).

3. If the orbit is elliptical, compute the Mean Anomaly, M. Then compute r, the heliocentric distance, and v, the true anomaly.

4. If the orbit is a parabola, or close to a parabola (the eccentricity is 1.0 or nearly 1.0), then the algorithms for elliptical orbits will break down. Then use another algorithm, presented below, to compute r, the heliocentric distance, and v, the true anomaly, for near-parabolic orbits.

5. When you know r and v, proceed as with the planets: compute first the heliocentric, then the geocentric, position.

6. If needed, precess the final position to the desired epoch, e.g. 2000.0

A quantity we'll encounter here is Gauss' gravitational constant, k. This constant links the Sun's mass with our time unit (the day) and the length unit (the astronomical unit). The EXACT value of Gauss' gravitational constant k is:

    k = 0.01720209895   (exactly!)

If the orbit is elliptical, and if the perihelion distance, q, is given
instead of the mean distance, a, we start by computing the mean distance a from
the perihelion distance q and the eccentricity e:

    a = q / (1 - e)

Now we compute the Mean Anomaly, M:

    M = (180_deg/pi) * (d - D) * k / (a ** 1.5)

    a ** 1.5 is most easily computed as: sqrt(a*a*a)

Now we know the Mean Anomaly, M. We proceed as for a planetary orbit by
computing E, the eccentric anomaly. Since comet and asteroid orbits often have
high eccentricities, we must use the iteration formula given earlier, and be
sure to iterate until we get convergence for the value of E.

The orbital period for a comet or an asteroid in elliptic orbit is (P in days):

    P = 2 * pi * (a ** 1.5) / k

If the comet's orbit is a parabola, the algorithm for elliptic orbits will break down: the semi-major axis and the orbital period will be infinite, and the Mean Anomaly will be zero. Then we must proceed in a different way. For a parabolic orbit we start by computing the quantities a, b and w (where a is not at all related to a for an elliptic orbit):

```
a = 1.5 * (d - D) * k / sqrt(2 * q*q*q)

b = sqrt( 1 + a*a )

w = cbrt(b + a) - cbrt(b - a)
```

cbrt is the Cubic Root function. Finally we compute the true anomaly, v, and the heliocentric distiance, r:

```
v = 2 * atan(w)
r = q * ( 1 + w*w )
```

From here we can proceed as usual.

Finally we have the case that's most common for newly discovered comets: the orbit isn't an exact parabola, but very nearly so. It's eccentricity is slightly below, or slightly above, one. The algorithm presented here can be used for eccentricities between about 0.98 and 1.02. If the eccentricity is smaller than 0.98 the elliptic algorithm should be used instead. No known comet has an eccentricity exceeding 1.02.

As for the purely parabolic orbit, we start by computing the time since perihelion in days, d - D, and the perihelion distance, q. We also need to know the eccentricity, e. Then we can proceed as:

```
a = 0.75 * (d - D) * k * sqrt( (1 + e) / (q*q*q) )
b = sqrt( 1 + a*a )
W = cbrt(b + a) - cbrt(b - a)
c = 1 + 1/(W*W)
f = (1 - e) / (1 + e)
g = f / (c*c)

a1 = (2/3) + (2/5) * W*W
a2 = (7/5) + (33/35) * W*W + (37/175) * W**4
a3 = W*W * ( 432/175) + (956/1125) * W*W + (84/1575) * W**4 )

w = W * ( 1 + g * c * ( a1 + a2*g + a3*g*g ) )

v = 2 * atan(w)
r = q * ( 1 + w*w ) / ( 1 + w*w * f )
```

This algorithm yields the true anomaly, v, and the heliocentric distance, r, for a nearly-parabolic orbit.

Now it's time for a practical example. Let's select two of the comets that were seen in the autumn of 1990: Comet Encke, a well-known periodic comet, and comet Levy, which was easily seen towards a dark sky in the autumn of 1990. When passing the inner solar system, the orbit of comet Levy was slightly hyperbolic.

According the the Handbook of the British Astronomical Association the orbital elements for comet Encke in 1990 are:

```
T = 1990 Oct 28.54502 TDT
e = 0.8502196
```

q = 0.3308858
w = 186.24444_deg
N = 334.04096_deg   1950.0
i = 11.93911_deg

The orbital elements for comet Levy are (BAA Circular 704):

T = 1990 Oct 24.6954 ET
e = 1.000270
q = 0.93858
w = 242.6797_deg
N = 138.6637_deg   1950.0
i = 131.5856_deg

Let's also choose another test date, when both these comets were visible:
1990 Aug 22, 0t UT, which yields a "day number" d = -3418.0

Now we compute the day numbers at perihelion for these two comets. We get for comet Encke:

D = -3350.45498   d - D = -67.54502

and for comet Levy:

D = -3354.3046   d - D = -63.6954

We'll continue by computing the Mean Anomaly for comet Encke:

M = -20.2751_deg = 339.7249_deg

The first approximation plus successive approximation for the Eccentric
anomaly, E, becomes (degrees):

E = 309.3811   293.5105   295.8474   295.9061   295.9061_deg ....

Here we clearly see the great need for iteration: the initial
approximation differs from the final value by 14 degrees. Finally we compute the
true anomaly, v, and heliocentric distance, r, for comet Encke:

v = 228.8837_deg
r = 1.3885

Now it's time for comet Levy: we'll compute the true anomaly, v, and the
heliocentric distance, r, for Levy in two different ways. First we'll pretend
that the orbit of Levy is an exact parabola. We get:

a = -1.2780823   b = 1.6228045   w = -0.7250189

v = -71.8856_deg
r = 1.431947

Then we repeat the computation but accounts for the fact that Levy's orbit
deviates slightly from a parabola. We get:

a = -1.2781686   b = 1.6228724   W = -0.7250566
c = 2.9022000   f = -1.3498E-4   g = -1.60258E-5
a1= 0.8769495   a2= 1.9540987   a3= 1.5495702
w = -0.7250270

v = -71.8863_deg
r = 1.432059

The difference is small in this case - only 0.0007 degrees or 2.5 arc seconds in true anomaly, and 0.000112 a.u. in heliocentric distance. Here it would have been sufficient to treat Levy's orbit as an exact parabola.

Now we know the true anomaly, v, and the heliocentric distance, r, for both Encke and Levy. Next we proceed by precessing N, the longitude of the ascending node, from 1950.0 to the "epoch of the day". Let's compute the precession angle from 1950.0 to 1990 Aug 22:

prec = 3.82394E-5_deg * (365.2422 * (1950.0 - 2000.0) - (-3418))
prec = -0.5676_deg

To precess from 1990 Aug 22 to 1950.0, we should add this angle to N. But now we want to do the opposite: precess from 1950.0 to 1990 Aug 22, therefore we must instead subtract this angle:

For comet Encke we get:

N = 334.04096_deg - (-0.5676_deg) = 334.60856_deg

and for comet Levy we get:

N = 138.6637_deg - (-0.5676_deg) = 139.2313_deg

Using this modified value for N we proceed just like for the planets. I won't repeat the details, but merely state some intermediate and final results:

Sun's position: x = -0.863890 y = +0.526123

Heliocentric:	Encke	Levy
x	+1.195087	+1.169908
y	+0.666455	-0.807922
z	+0.235663	+0.171375

Geoc., eclipt.:	Encke	Levy
x	+0.331197	+0.306018
y	+1.192579	-0.281799
z	+0.235663	+0.171375

Geoc., equat.:	Encke	Levy
x	+0.331197	+0.306018
y	+1.000414	-0.326716
z	+0.690619	+0.045133
RA	71.6824_deg	313.1264_deg
Decl	+33.2390_deg	+5.7572_deg
R	1.259950	0.449919

These positions are for the "epoch of the day". If you want positions for some standard epoch, e.g. 2000.0, these positions must be precessed to that epoch.

Finally some notes about computing the magnitude of a comet. To accurately predict a comet's magnitude is usually hard and sometimes impossible. It's fairly common that a magnitude prediction is off by 1-2 magnitudes or even more. For comet Levy the magnitude formula looked like this:

m = 4.0 * 5*log10(R) + 10*log10(r)

where R is the geocentric distance and r the heliocentric distance. The general case is:

m = G * 5*log10(R) + H*log10(r)

where H usually is around 10. If H is unknown, it's usually assumed to be 10. Each comet has it's own G and H.

Some comets have a different magnitude formula. One good example is comet Encke, where the magnitude formula looks like this:

m1 = 10.8 + 5*log10(R) + 3.55 * (r**1.8 - 1)

"m1" refers to the total magnitude of the comet. There is another cometary magnitude, "m2", which refers to the magnitude of the nucleus of the comet. The magnitude formula for Encke's m2 magnitude looks like this:

m2 = 14.5 + 5*log10(R) + 5*log10(r) + 0.03*FV

Here FV is the phase angle. This kind of magnitude formula looks very much like the magnitude formula of asteroids, for a very good reason: when a comet is far away form the Sun, no gases are evaporated from the surface of the comet. Then the comet has no tail (of course) and no coma, only a nucleus. Which means the comet then behaves much like an asteroid.

During the last few years it's become increasingly obvious that comets and asteroids often are similar kinds of solar-system objects. The asteroid (2060) Chiron has displayed cometary activity and is now also considered a comet. And in some cases comets that have "disappeared" have been re-discovered as asteroids! Apparently they "ran out of gas" and what remains of the former comet is only rock, i.e. an asteroid.