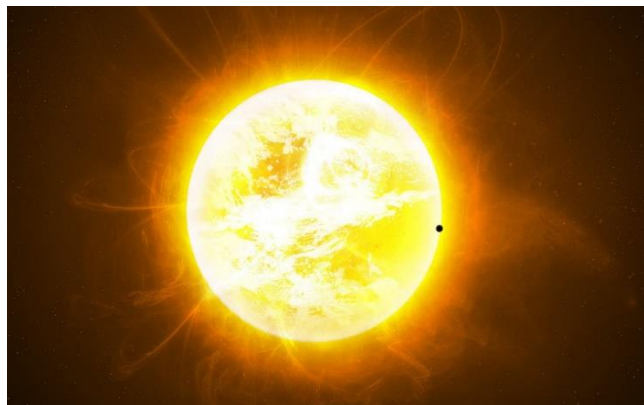




Теория принятия решений в астрономии

Системы координат

Гелиоцентрические

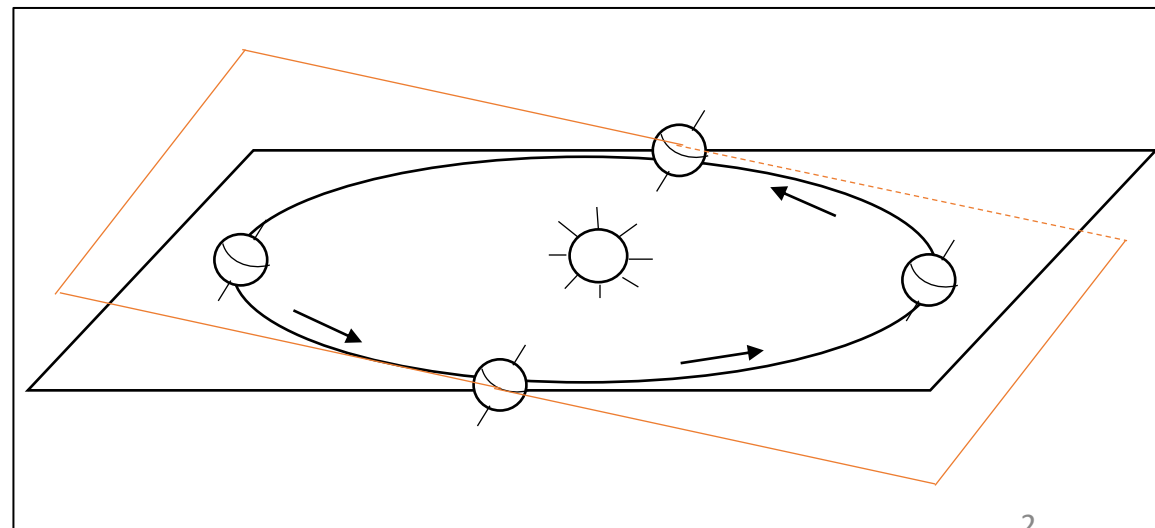


Геоцентрические



Эклиптические

Экваториальные





Модуль с константами APC_Const

```
const double pi      = 3.14159265358979324;  
const double pi2     = 2.0*pi;  
const double Rad     = pi / 180.0;  
const double Deg     = 180.0 / pi;  
const double Arcs    = 3600.0*180.0/pi;  
const double AU      = 149597870.0;  // Astronomical unit [km]  
  
const double c_light = 173.14;      // speed of light [AU/d]
```

Dd	DMS
15.5000	15 30 00.0
-8.15278	-8 9 10.0

Преобразование углов из градусов, минут и секунд
дуги в десятичное представление

```
double Ddd (int D, int M, double S)  
{  
    double sign;  
    if ( (D<0) || (M<0) || (S<0) ) sign = -1.0; else sign = 1.0;  
  
    return sign * ( fabs(D)+fabs(M)/60.0+fabs(S)/3600.0 );  
}
```

Вычисление градусов, минут и секунд дуги
по заданному значению

```
void DMS (double Dd, int& D, int& M, double& S)  
{  
    double x;  
    x = fabs(Dd);  D = int(x);  
    x = (x-D)*60.0; M = int(x);  S = (x-M)*60.0;  
    if (Dd<=0.0) { if (D!=0) D*=-1; else if (M!=0) M*=-1; else  
S*=-1.0; }  
}
```

Шестидесятеричная система счисления

Древние Шумеры

𐎶 1	𐎶𐎵 11	𐎶𐎵𐎶 21	𐎶𐎵𐎶𐎶 31	𐎶𐎵𐎶𐎶𐎶 41	𐎶𐎵𐎶𐎶𐎶𐎶 51
𐎶𐎶 2	𐎶𐎶𐎵 12	𐎶𐎶𐎶 22	𐎶𐎶𐎶𐎶 32	𐎶𐎶𐎶𐎶𐎶 42	𐎶𐎶𐎶𐎶𐎶𐎶 52
𐎶𐎶𐎶 3	𐎶𐎶𐎶𐎵 13	𐎶𐎶𐎶𐎶 23	𐎶𐎶𐎶𐎶𐎶 33	𐎶𐎶𐎶𐎶𐎶𐎶 43	𐎶𐎶𐎶𐎶𐎶𐎶𐎶 53
𐎶𐎶𐎶𐎶 4	𐎶𐎶𐎶𐎶𐎵 14	𐎶𐎶𐎶𐎶𐎶 24	𐎶𐎶𐎶𐎶𐎶𐎶 34	𐎶𐎶𐎶𐎶𐎶𐎶𐎶 44	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 54
𐎶𐎶𐎶𐎶𐎶 5	𐎶𐎶𐎶𐎶𐎶𐎵 15	𐎶𐎶𐎶𐎶𐎶𐎶 25	𐎶𐎶𐎶𐎶𐎶𐎶𐎶 35	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 45	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 55
𐎶𐎶𐎶𐎶𐎶𐎶 6	𐎶𐎶𐎶𐎶𐎶𐎶𐎵 16	𐎶𐎶𐎶𐎶𐎶𐎶𐎶 26	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 36	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 46	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 56
𐎶𐎶𐎶𐎶𐎶𐎶𐎶 7	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 17	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 27	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 37	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 47	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 57
𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 8	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 18	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 28	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 38	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 48	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 58
𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 9	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 19	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 29	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 39	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 49	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 59
𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 10	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 20	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 30	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 40	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 50	

Письмо царю, 2400 г. до н. э.



```

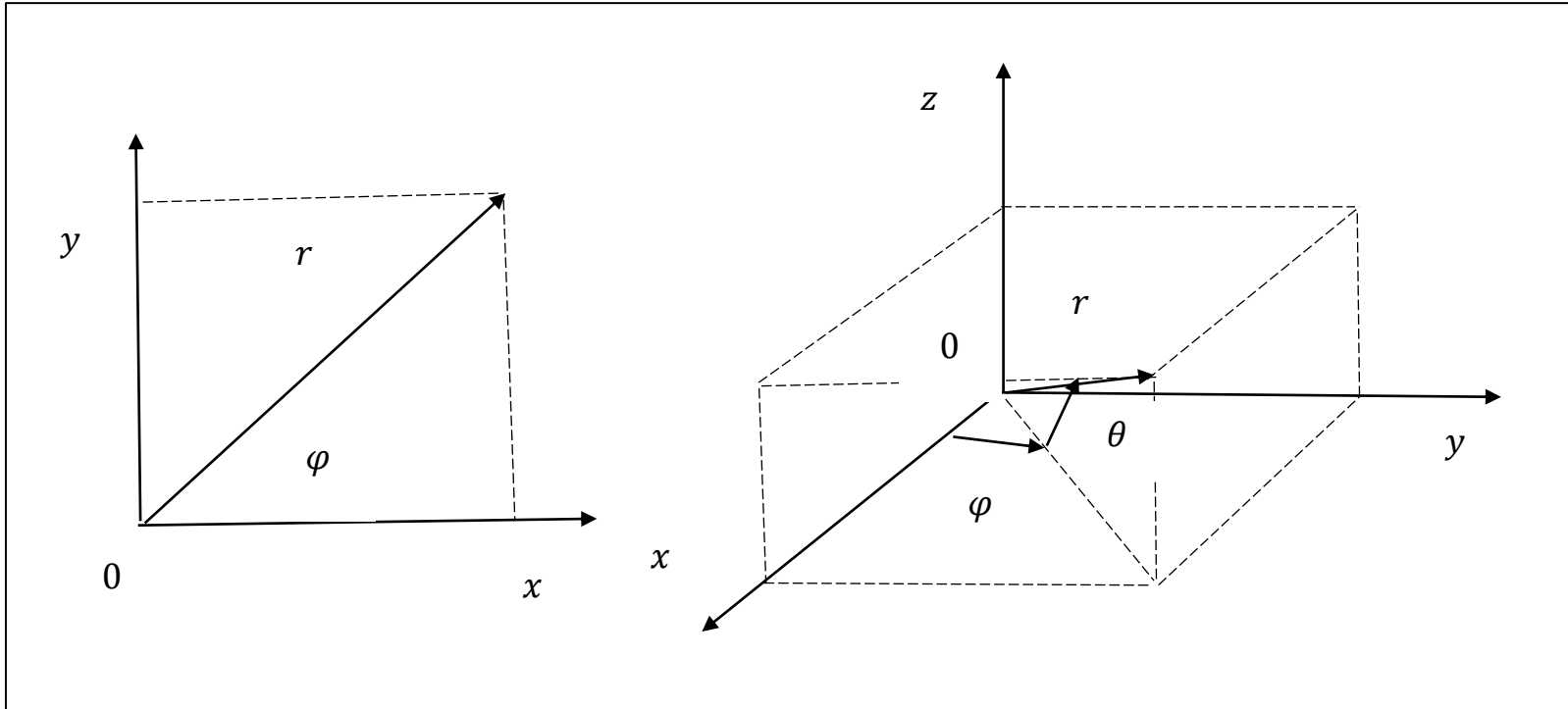
enum AngleFormat {
    Dd,    // decimal representation
    DMM,   // degrees and whole minutes of arc
    DMMm,  // degrees and minutes of arc in decimal representation
    DMMSS, // degrees, minutes of arc and whole seconds of arc
    DMMSSs // degrees, minutes, and seconds of arc in decimal representation
};

class Angle
{
public:
    // Constructor
    Angle (double alpha, AngleFormat Format=Dd);
    // Modifiers
    void Set (AngleFormat Format=Dd);
    // Angle output
    friend std::ostream& operator << (std::ostream& os, const Angle& alpha);
private:
    double    m_angle;
    AngleFormat m_Format;
};

```

APC_VetMat3D

Связь декартовых координат на плоскости и в пространстве



$$x = r \cos \theta \cos \varphi$$

$$y = r \cos \theta \sin \varphi$$

$$z = r \sin \theta$$

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\operatorname{tg} \varphi = \frac{y}{x}$$

$$\operatorname{tg} \theta = \frac{z}{\sqrt{x^2 + y^2}}$$

```

struct Polar {
    // Constructors
    Polar();
    Polar(double Az, double Elev, double R = 1.0);
    // Members
    double phi;    // azimuth of vector
    double theta;  // altitude of vector
    double r;      // norm of vector
};

class Mat3D
{
public:
    Mat3D (); // default constructor for null matrix
    // constructor for matrix from column vectors
    Mat3D ( const Vec3D& e_1, const Vec3D& e_2, const
Vec3D& e_3 );
    // component access
    friend Vec3D Col(const Mat3D& Mat, index Index);
    friend Vec3D Row(const Mat3D& Mat, index Index);
    // identity matrix
    friend Mat3D Id3D();

```

```

void Vec3D::CalcPolarAngles ()
{ // Length of projection in x-y-plane:
    const double rhoSqr = m_Vec[0] * m_Vec[0] + m_Vec[1] *
m_Vec[1];
    // Norm of vector
    m_r = sqrt ( rhoSqr + m_Vec[2] * m_Vec[2] );
    // Azimuth of vector
    if ( (m_Vec[0]==0.0) && (m_Vec[1]==0.0) )
        m_phi = 0.0;
    else
        m_phi = atan2 (m_Vec[1], m_Vec[0]);
    if ( m_phi < 0.0 ) m_phi += 2.0*pi;
    // Altitude of vector
    const double rho = sqrt ( rhoSqr );
    if ( (m_Vec[2]==0.0) && (rho==0.0) )
        m_theta = 0.0;
    else
        m_theta = atan2(m_Vec[2], rho);}

```

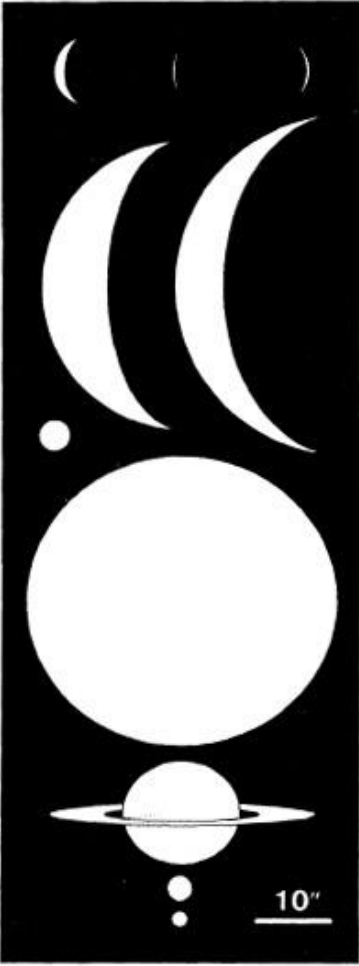
Матрицы поворотов

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{pmatrix}$$

$$R_y(\varphi) = \begin{pmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{pmatrix}$$

$$R_z(\varphi) = \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Эфемериды



	$\alpha_{2000.0}$ h m	$\delta_{2000.0}$ ° '	m	D	f	Видимость			Созвездие
				' "		$\varphi=45^\circ$ h	$\varphi=55^\circ$ h	$\varphi=65^\circ$ h	
Солнце	1.05 02 35	+15 06	-26.8	31 44					Овен
	16.05 03 32	+19 07	-26.8	31 38					Телец
	31.05 04 33	+21 56	-26.8	31 33					Телец
Меркурий	1.05 03 42	+22 14	+1.7	9.8	0.16	0.5	—	—	Телец
	11.05 03 37	+19 57	+4.8	11.8	0.01	—	—	—	Телец
	21.05 03 18	+15 54	+3.7	11.9	0.04	—	—	—	Овен
	31.05 03 14	+14 06	+1.6	10.2	0.19	—	—	—	Овен
Венера	1.05 05 24	+27 41	-4.5	35.1	0.30	4.2	5.0	7.1	Телец
	16.05 05 51	+27 23	-4.5	44.5	0.17	3.4	3.9	4.9	Телец
	31.05 05 42	+25 17	-4.1	54.8	0.04	1.5	1.6	—	Телец
Марс	1.05 01 47	+10 31	+1.3	3.9	0.99	—	—	—	Овен
	16.05 02 03	+14 25	+1.3	3.9	0.99	—	—	—	Овен
	31.05 03 14	+17 46	+1.4	4.0	0.99	—	—	—	Овен
Юпитер	1.05 19 16	-22 12	-2.4	41.5	0.99	4.7	3.4	—	Стрелец
	16.05 19 16	-22 15	-2.5	43.4	0.99	5.4	3.9	—	Стрелец
	31.05 19 13	-22 23	-2.6	45.1	1.00	6.2	4.6	—	Стрелец
Сатурн	1.05 00 13	-00 49	+1.3	16.1	1.00	—	—	—	Рыбы
	16.05 00 19	-00 17	+1.3	16.3	1.00	1.0	—	—	Рыбы
	31.05 00 23	+00 11	+1.2	16.7	1.00	1.9	—	—	Рыбы
Уран	16.05 20 28	-19 04	+5.7	3.6	1.00	4.2	2.8	—	Козерог
Нептун	16.05 19 59	-20 08	+7.9	2.3	1.00	4.2	2.6	—	Стрелец
Плутон	16.05 16 01	-07 02	+13.7	0.14	1.00				Змееносец

В таблице приведены эфемериды Солнца и планет: прямое восхождение и склонение (на эпоху 2000.0 года), видимая звездная величина, диаметр, фаза. Продолжительность видимости планет рассчитана программой, составленной О. С. Угольниковым, с учетом зависимостей неравномерной яркости сумеречного неба и предельной видимой звездной величины от глубины погружения Солнца под горизонт. Для Урана и Нептуна рассчитана продолжительность видимости в телескоп с диаметром объектива 10 см при увеличении 30 \times .

Юлианский период от 01 января 4713 г. до н. э.

Модифицированная юлианская дата

$$MJD = JD - 2400000,5$$

отсчитывается от полуночи 17 ноября 1858 г.

APC_Time

```
double Mjd ( int Year, int Month, int Day,           // Mjd: Modified Julian Date from calendar date and time
             int Hour, int Min, double Sec )         // Input:
{ // Variables                                       // Year    Calendar date components
    long  MjdMidnight;                             // Month
    double FracOfDay;                              // Day
    int   b;                                        // Hour    Time components (optional)
    if (Month<=2) { Month+=12; --Year;}             // Min
    if ( (10000L*Year+100L*Month+Day) <= 15821004L ) // Sec
        b = -2 + ((Year+4716)/4) - 1179;    // Julian calendar // <return>: Modified Julian Date
    else
        b = (Year/400)-(Year/100)+(Year/4); // Gregorian calendar
    MjdMidnight = 365L*Year - 679004L + b +
    int(30.6001*(Month+1)) + Day;
    FracOfDay  = Ddd(Hour,Min,Sec) / 24.0;
    return MjdMidnight + FracOfDay;}
```

Григорианская реформа – после 4 октября 1582 г. (JD 2299159,5) следовала дата 15 октября 1582 года (JD 2299160,5).

Последние годы столетий, номера которых не делятся на 400 не считаются високосными

Средняя длина года по **григорианскому** календарю $365 + \frac{1}{4} - \frac{1}{100} + \frac{1}{400} = 365,2425$

```

void CalDat ( double Mjd,
              int& Year, int& Month, int& Day, double & Hour )
{ // Variables
  long  a,b,c,d,e,f;
  double  FracOfDay;
  // Convert Julian day number to calendar date
  a = long(Mjd+2400001.0);
  if ( a < 2299161 ) { // Julian calendar
    b = 0;
    c = a + 1524; }
  else { // Gregorian calendar
    b = long((a-1867216.25)/36524.25);
    c = a + b - (b/4) + 1525; }
  d  = long ( (c-122.1)/365.25 );
  e  = 365*d + d/4;
  f  = long ( (c-e)/30.6001 );
  Day  = c - e - int(30.6001*f);
  Month = f - 1 - 12*(f/14);
  Year  = d - 4715 - ((7+Month)/10);
  FracOfDay = Mjd - floor(Mjd);
  Hour = 24.0*FracOfDay;}

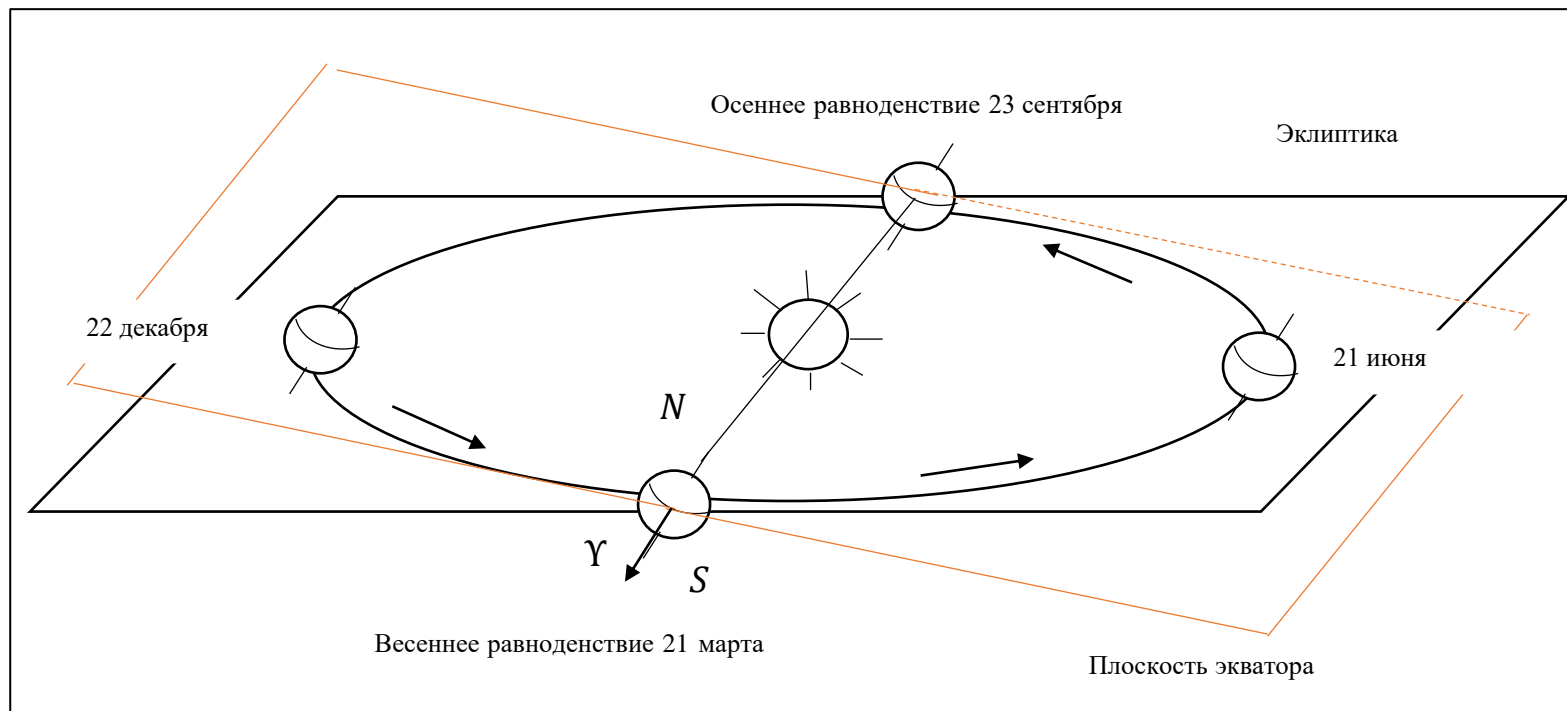
```

```

void CalDat ( double Mjd,
              int& Year, int& Month, int& Day,
              int& Hour, int& Min, double& Sec )
{
  //
  // Variables
  //
  double Hours;
  CalDat (Mjd, Year, Month, Day, Hours);
  DMS (Hours, Hour, Min, Sec);
}

```

Эклиптические и экваториальные координаты



Произвольная **эклиптическая** точка (x, y, z) в **экваториальной** системе будет иметь координаты:

$$x' = x$$

$$y' = y \cos \varepsilon - z \sin \varepsilon$$

$$z' = y \sin \varepsilon + z \cos \varepsilon$$

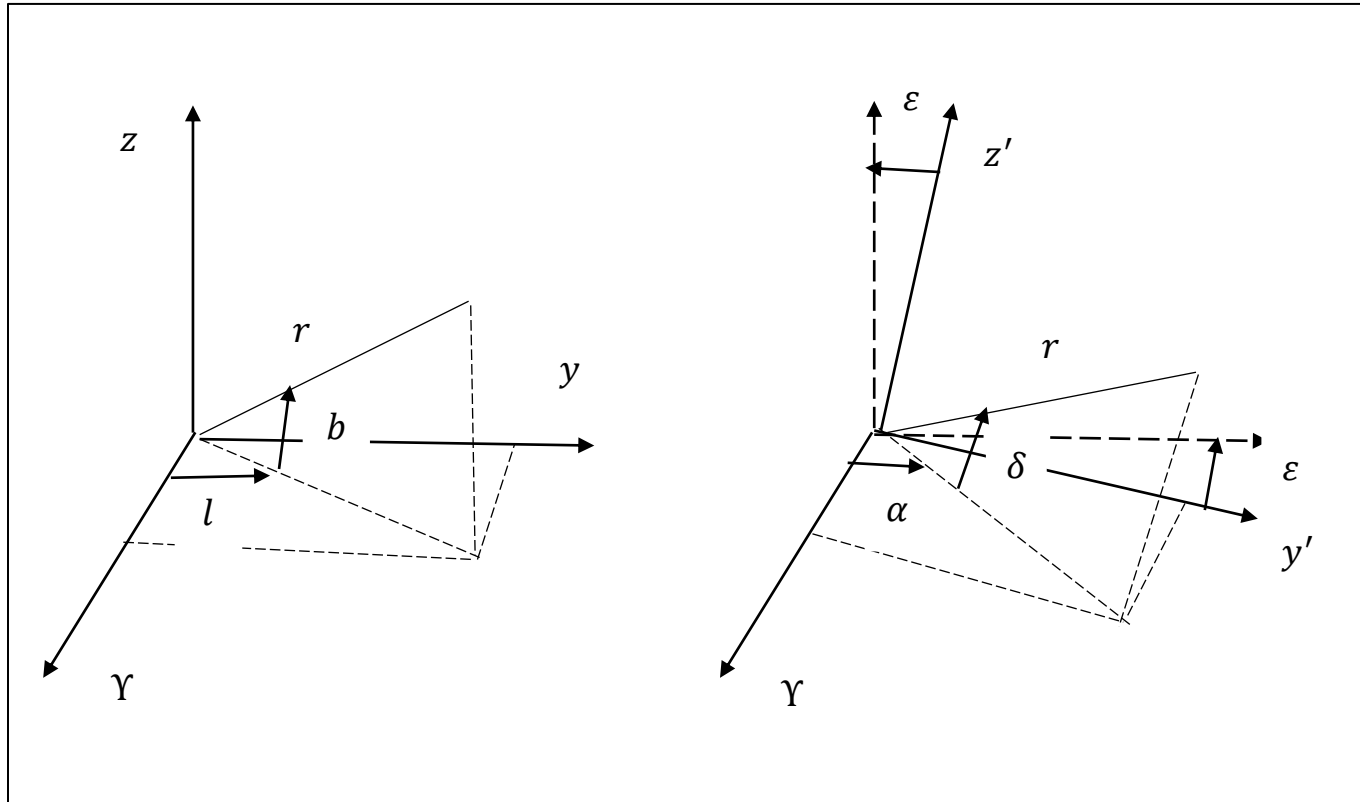
Обратные соотношения:

$$x = x'$$

$$y = y' \cos \varepsilon + z' \sin \varepsilon$$

$$z = -y' \sin \varepsilon + z' \cos \varepsilon$$

Связь эклиптических, экваториальных и полярных координат



l — эклиптическая **долгота**,
 b — эклиптическая **широта**.

α — **прямое восхождение**,
 δ — **склонение**.

$$x = r \cos b \cos l$$

$$y = r \cos b \sin l$$

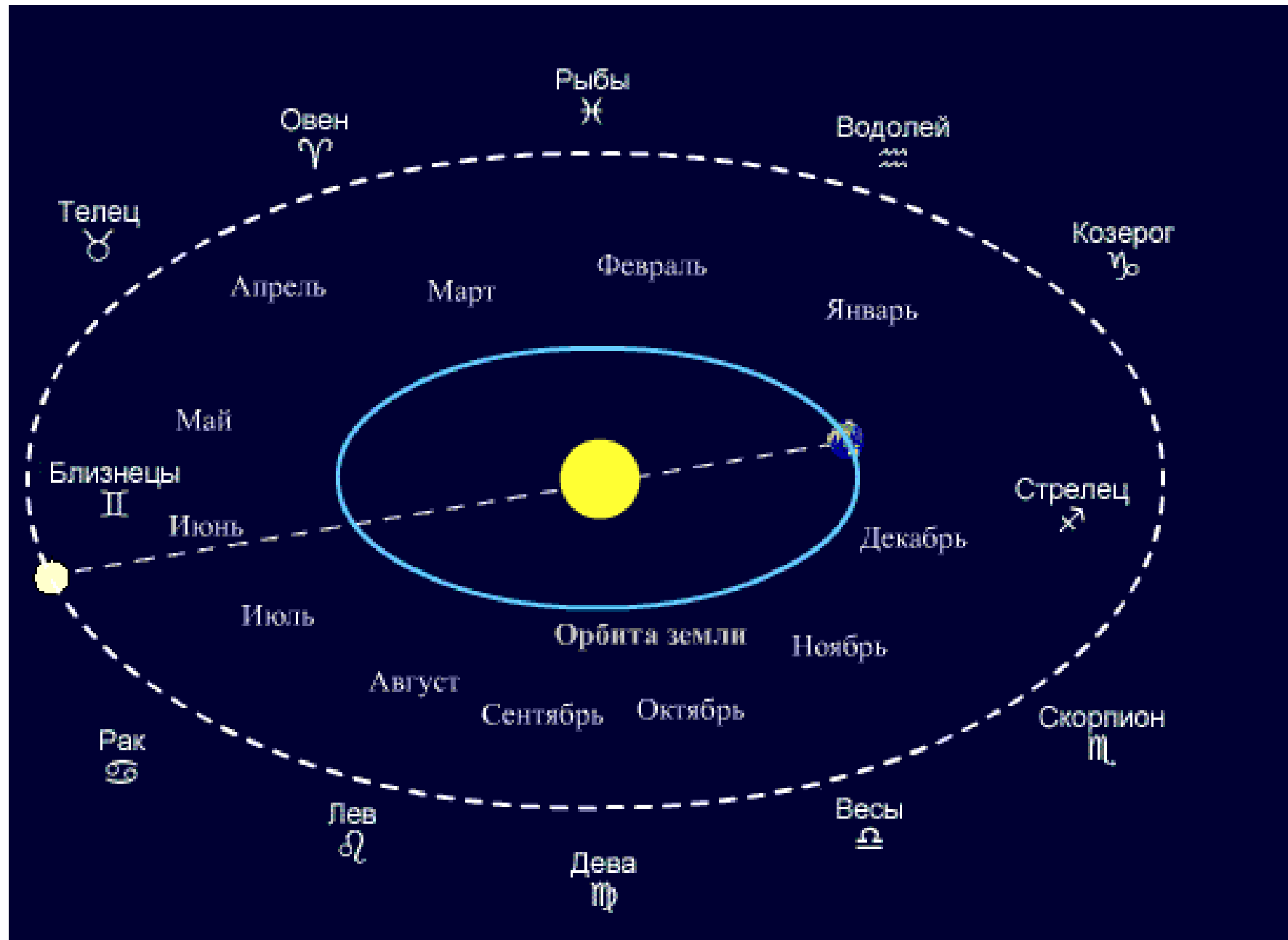
$$z = r \sin b$$

$$x' = r \cos \delta \cos \alpha$$

$$y' = r \cos \delta \sin \alpha$$

$$z' = r \sin \delta$$

Зодиакальный пояс



Эклиптическая система координат (x, y, z)

Эклиптическая долгота знаков Зодиака

Овен	00° - 30 °	(21 март - 20 апреля)
Телец	30° - 60 °	(21 апреля - 20 мая)
Близнецы	60° - 90 °	(21 мая - 21 июня)
Рак	90° - 120 °	(22 июня - 22 июля)
Лев	120° -150 °	(23 июля - 23 августа)
Дева	150° -180 °	(24 августа - 23 сентября)
Весы	180° -210 °	(24 сентября - 23 октября)
Скорпион	210° -240 °	(24 октября - 22 ноября)
Стрелец	240° -270 °	(23 ноября - 21 декабря)
Козерог	270° -300 °	(22 декабря - 21 января)
Водолей	300° -330 °	(21 января - 19 февраля)
Рыбы	330° -360 °	(20 февраля - 20 марта)

Первая координата - l (долгота).

Вторая координата - b (широта).

Эклиптическая геоцентрическая.

Эклиптическая гелиоцентрическая.

Экваториальная система координат (x', y', z')

Первая координата - α (**прямое восхождение**)

Прямое восхождение равно длине дуги небесного экватора от точки весеннего равноденствия до круга склонения светила.

- Точка весеннего равноденствия имеет прямое восхождение 0^h ;
- Точка летнего солнцестояния имеет прямое восхождение 6^h ;
- Точка осеннего равноденствия имеет прямое восхождение 12^h ;
- Точка зимнего солнцестояния имеет прямое восхождение 18^h .

Вторая координата - δ (**склонение**)

Склонение равно угловому расстоянию на небесной сфере от плоскости небесного экватора до светила, причём оно положительно для северной полусферы и отрицательно для южной.

- Любая точка небесного экватора имеет склонение 0° ;
- Склонение северного полюса мира равно $+90^\circ$;
- Склонение южного полюса мира равно -90° .

Преобразование между эклиптическими и экваториальными координатами

$$r = R_x(\varepsilon)r'$$

$$r' = R_x^T(\varepsilon)r = R_x(-\varepsilon)r$$

Изменение наклона эклиптики

$$\varepsilon = 23^\circ,43929111 - 46'',8150T - 0'',00059T^2 + 0'',001813T^3$$

T – число юлианских столетий, отделяющих эпоху от полудня 1 января 2000 г.

Юлианские эпохи – J1900 (0,5 января 1900, JD 2415020), J2000(полдень января 2000, JD 2451545).

Величину T для данной эпохи можно вычислить по юлианской дате:

$$T = (JD\ 2451545)/36525$$

Модуль APC_Spheric

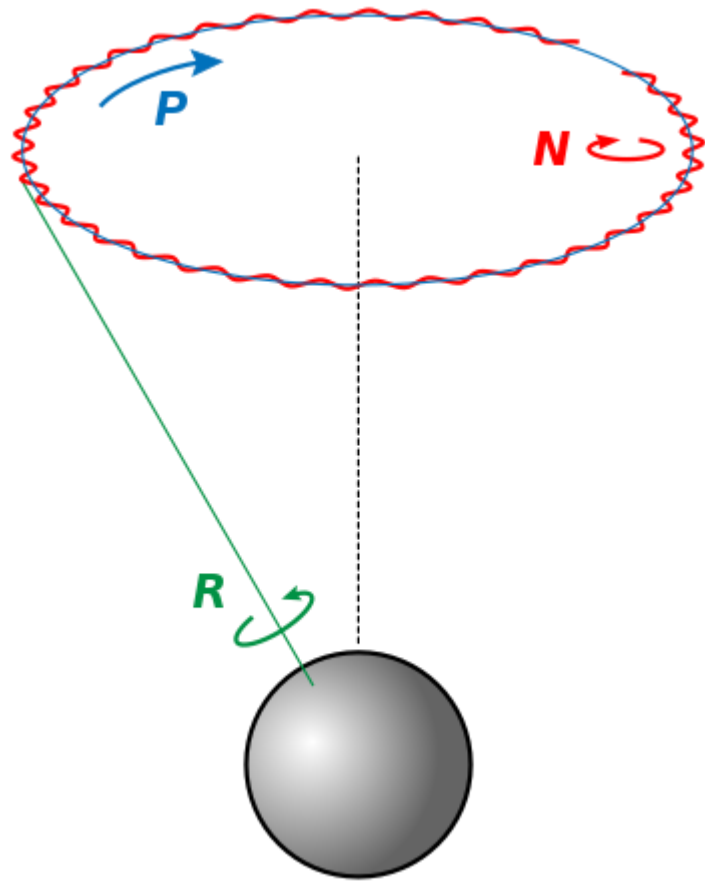
Transformation of equatorial to ecliptical coordinates

```
Mat3D Equ2EclMatrix (double T)
{ // Constants
  const double
    eps = ( 23.43929111-(46.8150+(0.00059-
0.001813*T)*T)*T/3600.0 ) * Rad;
  return R_x(eps);}
```

Transformation of ecliptical to equatorial coordinates

```
Mat3D Ecl2EquMatrix (double T)
{
  return Transp(Equ2EclMatrix(T));
}
```

Прецессия



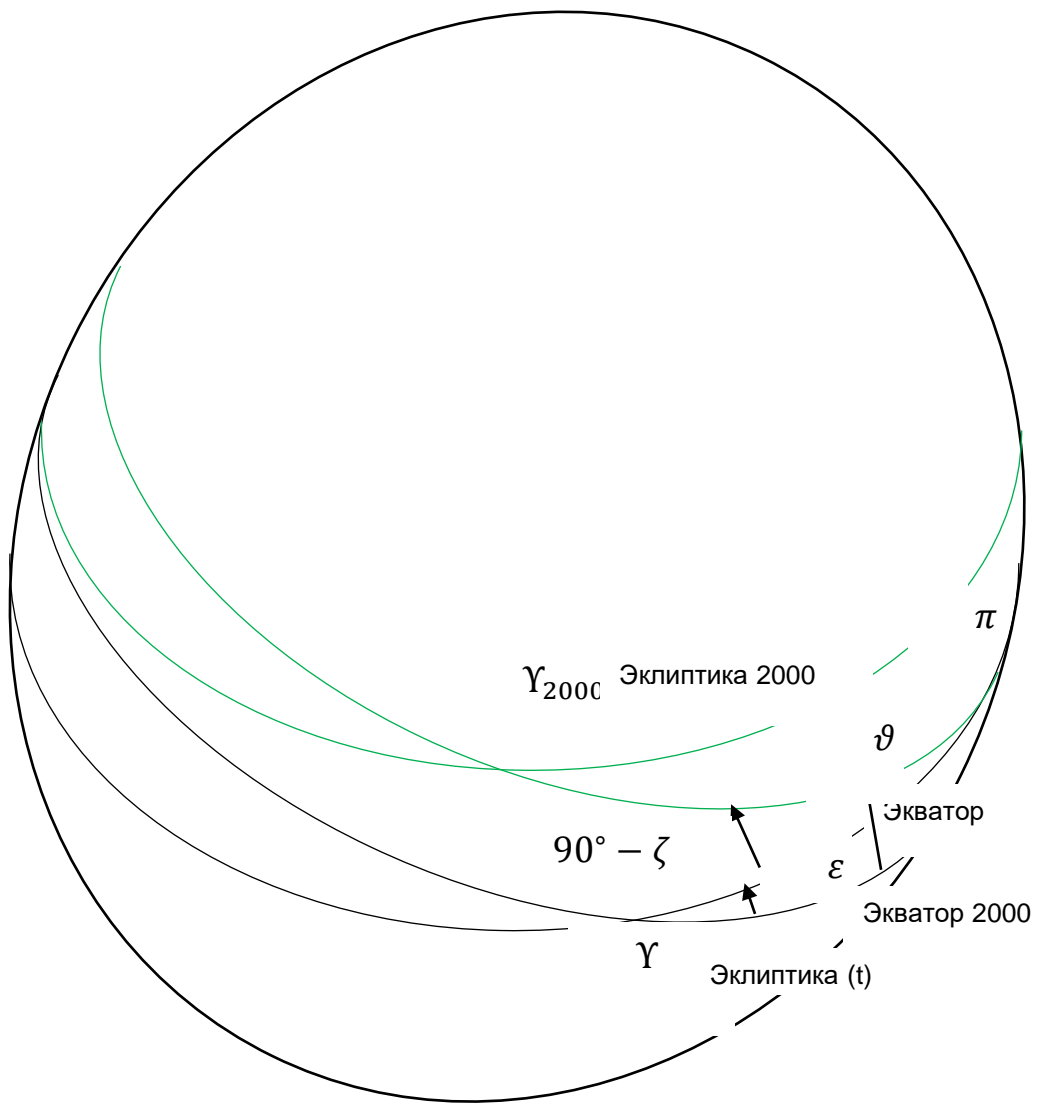
R — вращение, P - прецессия, N — нутация.



Гиппарх Никейский (190 – 125 г. до н. э.)

Наиболее распространенные эпохи:

- эпоха текущей даты;
- эпоха $J2000$; $J2000$ – 1,5 января 2000 = $JD2451545,0$
- эпоха $B1950$. $B1950$ – префикс В означает начало бесселева года (январь 1950 = $JD2433282,423$).



Влияние прецессии на взаиморасположение эклиптики, экватора и точки весеннего равноденствия

ε – угол между экватором и эклиптикой.

π – угол между эклиптикой и эклиптикой 2000.

ϑ – угол между экватором 2000 и экватором.

$(90^\circ - \zeta)$ – угол между экватором 2000 и эклиптикой.

Модуль APC_PrecNut.

Рассмотрим положение эклиптики в моменты времени T и $T + T_0$. Угол между этими плоскостями $\pi = p_i$.

Π – угол между осью x' и направлением на точку весеннего равноденствия Y_0 эпохи T_0 (ось x_0). $\Pi = \Pi_i$

Λ – угол между осью x'' и направлением на точку весеннего равноденствия эпохи $T_0 + T$ (ось x).

p – прецессия по долготе, $p = \Pi + \Lambda$. $p = p_a$

Mat3D **PrecMatrix_Ecl** (double T1, double T2)

```
{ // Constants
```

```
const double dT = T2-T1;
```

```
// Variables
```

```
double Pi, pi, p_a;
```

```
Pi = 174.876383889*Rad +
```

```
  ( ((3289.4789+0.60622*T1)*T1) +  
    ((-869.8089-0.50491*T1) + 0.03536*dT)*dT )/Arcs;
```

```
pi = ( (47.0029-(0.06603-0.000598*T1)*T1)+  
      ((-0.03302+0.000598*T1)+0.000060*dT)*dT )*dT/Arcs;
```

```
p_a = ( (5029.0966+(2.22226-0.000042*T1)*T1)+  
      ((1.11113-0.000042*T1)-0.000006*dT)*dT )*dT/Arcs;
```

```
return R_z(-(Pi+p_a)) * R_x(pi) * R_z(Pi);}
```

В экваториальных координатах углам π , Π и Λ соответствуют $90^\circ - \zeta$, ϑ и $90^\circ + z$.

$$zeta = \zeta$$

$$theta = \vartheta$$

Mat3D **PrecMatrix_Equ** (double T1, double T2)

```
{ // Constants
```

```
const double dT = T2-T1;
```

```
// Variables
```

```
double zeta,z,theta;
```

```
zeta = ( (2306.2181+(1.39656-0.000139*T1)*T1)+  
        ((0.30188-0.000344*T1)+0.017998*dT)*dT )*dT/Arcs;
```

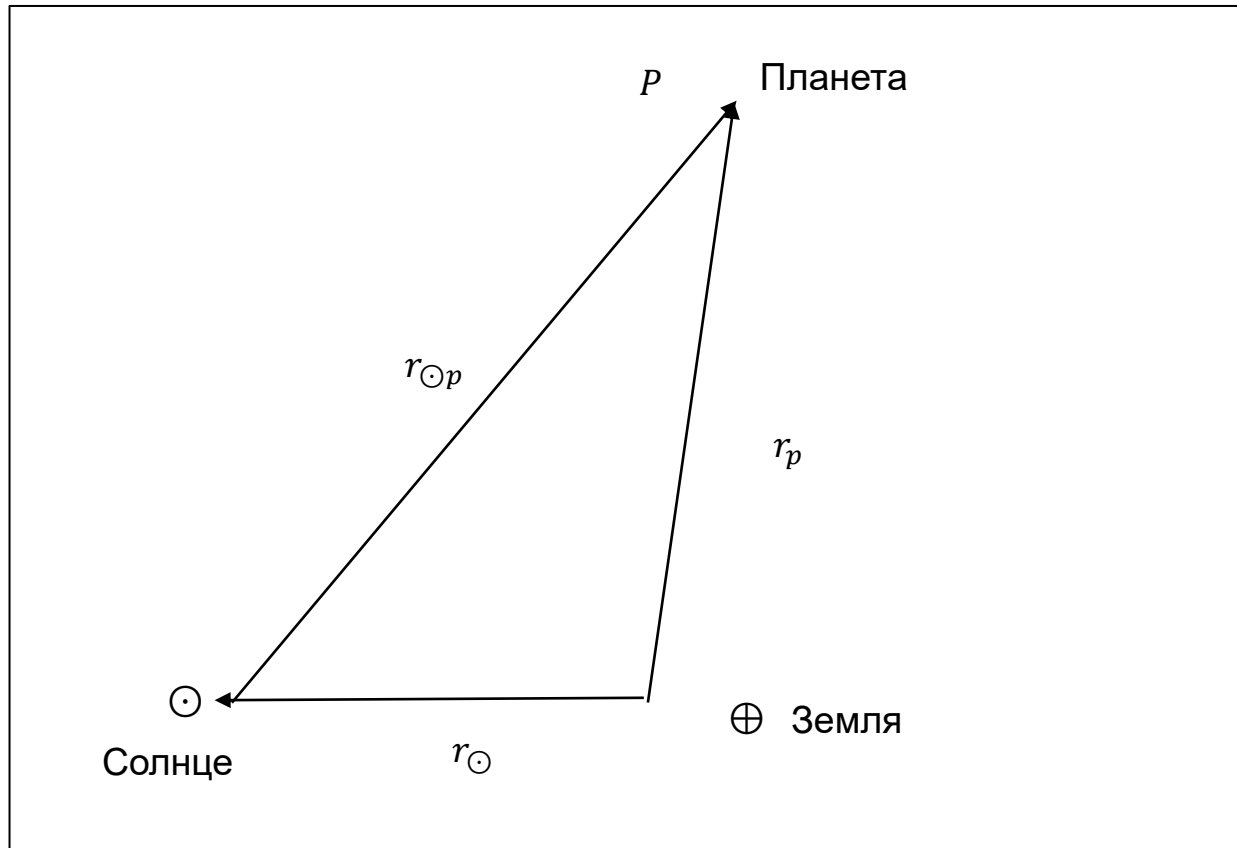
```
z = zeta + (  
(0.79280+0.000411*T1)+0.000205*dT)*dT*dT/Arcs;
```

```
theta = ( (2004.3109-(0.85330+0.000217*T1)*T1)-  
        ((0.42665+0.000217*T1)+0.041833*dT)*dT )*dT/Arcs;
```

```
return R_z(-z) * R_y(theta) * R_z(-zeta);}
```

Геоцентрические координаты и солнечная орбита

Переход от гелиоцентрических координат (отнесенных к центру Солнца) к геоцентрическим (отнесенным к центру Земли).



$$r_p = r_{\odot p} + r_{\odot}$$

$$r_{\odot p} = r_p - r_{\odot}$$

$r_{\odot p}$ – гелиоцентрический и r_p – геоцентрический радиус-векторы точки P .
 r_{\odot} – геоцентрический радиус вектор солнца.

$$x_p = x_{\odot p} + x_{\odot}$$

$$x_{\odot p} = x_p - x_{\odot}$$

$$y_p = y_{\odot p} + y_{\odot}$$

$$y_{\odot p} = y_p - y_{\odot}$$

$$z_p = z_{\odot p} + z_{\odot}$$

$$z_{\odot p} = z_p - z_{\odot}$$

$$r_{\odot} = \begin{pmatrix} x_{\odot} \\ y_{\odot} \\ z_{\odot} \end{pmatrix} = \begin{pmatrix} R \cos B \cos L \\ R \cos B \sin L \\ R \sin B \end{pmatrix}$$

Vec3D **SunPos** (double T)

Основная программа Coso

```
class Position
{ public:
    void Input(); // Запрос параметров
    void SetOrigin(enOrigin Origin);
    void SetRefSys(enRefSys RefSys);
    void SetEquinox(double T_Equinox);
    void Print();
private:
    Vec3D    m_R;      // Радиус-вектор
    enOrigin m_Origin;  // Начало координат
    enRefSys m_RefSys;  // Система координат
    double   m_TEquinox; // Равноденствие (в столетиях от J2000)
    double   m_MjdEpoch; // Эпоха (Модифицированная юлианская дата)};
```

Пример: выбирается экваториальные координаты точки весеннего равноденствия, заданные в полярной форме на эпоху 1950.0. Расстояние равно 1 а. е.

Проведем циклическое изменение команд

Сначала выберем **экваториальные** координаты точки весеннего равноденствия, заданные в **полярной** форме в эпоху 1950.0, расстояние 1 а. е.

```
COCO: coordinate conversions
(c) 1999 Oliver Montenbruck, Thomas Pfleger

New input:

Reference system (e=ecliptic,a=equator) ... a
Format (c=cartesian,p=polar) ... p
Coordinates (RA [h m s] Dec [o ' " ] R) ... 0 0 0.0 0 0 0.0 1.0
Equinox (yyyy.y) ... 1950.0
Origin (h=heliocentric,g=geocentric) ... g
Epoch (yyyy mm dd hh.h) ... 1989 1 1 0.0

Geocentric equatorial coordinates
(Equinox J1950.0, Epoch 1989/01/01 00.0)

(x,y,z) = ( 1.00000000, 0.00000000, 0.00000000)

      h m s           o ' "
RA = 0 00 00.00   Dec = + 0 00 00.0   R = 1.00000000

Enter command (?=Help) ... █
```

Программа ожидает команды преобразования координат.

а – преобразование в экваториальные координаты;

е – преобразование в эклиптические координаты;

р – прецессия (выбор эпохи);

g – преобразование в геоцентрические координаты;

h – преобразование в гелиоцентрические координаты;

X – ВЫХОД.

Рассчитаем координаты точки весеннего равноденствия 1950, отнесенные к новой эпохе 2000.0.

```
Enter command (?=Help) ... p
New equinox (yyyy.y) ... 2000.0

Geocentric equatorial coordinates
(Equinox J2000.0, Epoch 1989/01/01 00.0)

(x,y,z) = ( 0.99992571, 0.01117889, 0.00485898)

      h m s           o ' "
RA = 0 02 33.73    Dec = + 0 16 42.2    R = 1.00000000

Enter command (?=Help) ...
```

Преобразуем экваториальные координаты в эклиптические.

```
Enter command (?=Help) ... p
New equinox (yyyy.y) ... 2000.0

Geocentric equatorial coordinates
(Equinox J2000.0, Epoch 1989/01/01 00.0)

(x,y,z) = ( 0.99992571, 0.01117889, 0.00485898)

      h m s           o ' "
RA = 0 02 33.73    Dec = + 0 16 42.2    R = 1.00000000

Enter command (?=Help) ... e

Geocentric ecliptic coordinates
(Equinox J2000.0, Epoch 1989/01/01 00.0)

(x,y,z) = ( 0.99992571, 0.01218922, 0.00001132)

      o ' "           o ' "
L = 0 41 54.27    B = + 0 00 02.3    R = 1.00000000

Enter command (?=Help) ...
```

Преобразуем координаты из геоцентрических в гелиоцентрические на заданную эпоху, используя команду h.

```
Enter command (?=Help) ... h
```

```
Heliocentric ecliptic coordinates  
(Equinox J2000.0, Epoch 1989/01/01 00.0)
```

```
(x,y,z) = ( 0.81725247, 0.97838164, 0.00003597)
```

```
      o ' "  
L = 50 07 39.50    B = + 0 00 05.8    R = 1.27480674
```

Отменим преобразование в эклиптические координаты, выбрав a.

```
Enter command (?=Help) ... a
```

```
Enter command (?=Help) ... a
```

```
Heliocentric equatorial coordinates  
(Equinox J2000.0, Epoch 1989/01/01 00.0)
```

```
(x,y,z) = ( 0.81725247, 0.89763329, 0.38921086)
```

```
      h m s      o ' "  
RA = 3 10 44.07    Dec = +17 46 36.5    R = 1.27480674
```

Получили гелиоцентрические
эклиптические координаты.

Пересчитаем координаты к исходной эпохе 1950.0

```
Enter command (?=Help) ... p
New equinox (yyyy.y) ... 1950.0
```

```
Heliocentric equatorial coordinates
(Equinox J1950.0, Epoch 1989/01/01 00.0)
```

```
(x,y,z) = ( 0.82911747, 0.88843066, 0.38521087)
```

```
      h m s           o ' "
RA = 3 07 54.68    Dec = +17 35 17.2    R = 1.27480674
```

Преобразуем в **геоцентрические**.

```
Enter command (?=Help) ...
```

```
Enter command (?=Help) ... g
```

```
Geocentric equatorial coordinates
(Equinox J1950.0, Epoch 1989/01/01 00.0)
```

```
(x,y,z) = ( 1.00000000, -0.00000000, 0.00000000)
```

```
      h m s           o ' "
RA = 24 00 00.00    Dec = + 0 00 00.0    R = 1.00000000
```

Сравнение: не удалось войти дважды в одну реку

```
COCO: coordinate conversions
(c) 1999 Oliver Montenbruck, Thomas Pfleger
```

New input:

```
Reference system (e=ecliptic,a=equator) ... a
Format (c=cartesian,p=polar)           ... p
Coordinates (RA [h m s] Dec [o ' "] R) ... 0 0 0.0 0 0 0.0 1.0
Equinox (yyyy.y)                       ... 1950.0
Origin (h=heliocentric,g=geocentric)    ... g
Epoch (yyyy mm dd hh.h)                ... 1989 1 1 0.0
```

Enter command (?=Help) ... g

Geocentric equatorial coordinates
(Equinox J1950.0, Epoch 1989/01/01 00.0)

(x,y,z) = (1.00000000, 0.00000000, 0.00000000)

```
      h m s      o ' "
RA = 0 00 00.00  Dec = + 0 00 00.0  R = 1.00000000
```

Enter command (?=Help) ... █

Geocentric equatorial coordinates
(Equinox J1950.0, Epoch 1989/01/01 00.0)

(x,y,z) = (1.00000000, -0.00000000, 0.00000000)

```
      h m s      o ' "
RA = 24 00 00.00  Dec = + 0 00 00.0  R = 1.00000000
```

Enter command (?=Help) ...

