

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по практической работе №5
по дисциплине «Теория принятия решений»
Тема: Преобразование координат
Вариант 8

Студент гр. 8383

Киреев К.А.

Преподаватель

Попова Е.В.

Санкт-Петербург

2022

Выполнение работы

Программа

Были скачаны файлы программы, устранены баги и создано приложение из клиентских модулей. Зафиксированы изменения в коде.

- Не объявлены методы R_x , R_y и R_z в заголовочных файлах *APC_PrecNut.h* и *APC_Spheric.h*. Изменения представлены на рис. 1.

```
#include "APC_VecMat3D.h"

Mat3D R_x(double RotAngle);
Mat3D R_y(double RotAngle);
Mat3D R_z(double RotAngle);
```

Рисунок 1 – Объявление методов

- Не объявлено пространство имен *std* в *GNU_iomanip.h*.

```
#include <iomanip>
#include <iostream>
using namespace std;

namespace{
ostream& left (ostream& os){os.setf( fmtfl: ios::left , mask: ios::adjustfield); return os;};
ostream& right(ostream& os){os.setf( fmtfl: ios::right, mask: ios::adjustfield); return os;};
ostream& fixed(ostream& os){os.setf( fmtfl: ios::fixed, mask: ios::floatfield); return os;};
ostream& showpos (ostream& os){os.setf( fmtfl: ios::showpos); return os;};
ostream& noshowpos(ostream& os){os.unsetf( mask: ios::showpos); return os;};
}
```

Рисунок 2 – Пространство имен

- Неверный тип возвращаемого значения функцией *main*

Изменим возвращаемый функцией *void* тип в файле *Coso.cpp*

```
//-----
int main() {

    //
    // Variables
    //
```

Рисунок 3 – Тип возвращаемого значения

Циклические преобразования

Проведены циклические преобразования координат. Выбраны Reference system X, Format Y, Coordinates Z, Equinox K, Origin L, Epoch M. Эксперимент повторялся, пока не появилась погрешность входных данных.

Выбраны экваториальные координаты точки весеннего равноденствия, заданные в полярной форме в эпоху 1900.0, расстояние 1 а.е.

```
Reference system (e=ecliptic,a=equator) ...a
Format (c=cartesian,p=polar) ...p
Coordinates (RA [h m s] De
c [o ' "] R) ...0 0 0.0 0 0 0.0 1.0
Equinox (yyyy.y) ...1900.0
Origin (h=heliocentric,g=geocentric) ...g
Epoch (y
yyy mm dd hh.h) ...1993 1 1 0.0

Geocentric equatorial coordinates
(Equinox J2e+03, Epoch 1993/01/01 0.002)

(x,y,z) = 1, 0, 0

      h m s      o ' "
RA = 0 00 0.001  Dec = 0 00 0.01  R = 1
```

Рассчитаны координаты точки весеннего равноденствия 1900.0, отнесенные к новой эпохе 2000.0:

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

Geocentric equatorial coordinates
(Equinox J2e+03, Epoch 1993/01/01 0.002)

(x,y,z) = 0.99970291, 0.022352737,0.0097182792

      h m s      o ' "
RA = 0 05 007.4  Dec = 0 33 2e+01  R = 1
```

Преобразование экваториальных координат в эклиптические:

```

Enter command (?=Help) ...e

Geocentric ecliptic coordinates
(Equinox J2e+03, Epoch 1993/01/01 0.002)

(x,y,z) = 0.99970291, 0.024373944, 2.4938826e-05

      0   '   "           0   '   "
L = 1 23 00048   B = 0 00 0005   R = 1

```

Преобразование координат из геоцентрических в гелиоцентрические на заданную эпоху:

```

Enter command (?=Help) ...h

Heliocentric ecliptic coordinates
(Equinox J2e+03, Epoch 1993/01/01 0.002)

(x,y,z) = 0.81740473, 0.99063738, 3.8948038e-05

      0   '   "           0   '   "
L = 50 28 00023   B = 0 00 0006   R = 1.2843336

```

Отмена преобразования в эклиптические координаты:

```

Enter command (?=Help) ...a

Heliocentric equatorial coordinates
(Equinox J2e+03, Epoch 1993/01/01 0.002)

(x,y,z) = 0.81740473, 0.90887653, 0.39408865

      h   m   s           0   '   "
RA = 3 12 00008   Dec = 17 52 0009   R = 1.2843336

```

Пересчет координат к исходной эпохе 1900.0:

```

Enter command (?=Help) ...p
New equinox (yyyy.y) ...1900.0

Heliocentric equatorial coordinates
(Equinox J2e+03, Epoch 1993/01/01 0.002)

(x,y,z) = 0.84130763, 0.8903354, 0.38602752

      h m s          o ' "
RA = 3 06 00029    Dec = 17 29 3e+01    R = 1.2843336

```

Преобразование в геоцентрические координаты:

```

Enter command (?=Help) ...g

Geocentric equatorial coordinates
(Equinox J2e+03, Epoch 1993/01/01 0.002)

(x,y,z) = 1, -2.0521629e-10, 3.7410053e-10

      h m s          o ' "
RA = 24 00 0.001    Dec = 0 00 0.01    R = 1

```

Сравнение: значения не совпадают.

Испытания

После получения входных данных с погрешностью, была заполнена таблица 1 и проведено 30 испытаний с варьированием параметра прямое восхождение (минуты). Программа представлена на рис. 4.

Таблица 1

X	Y	Z	K	L	M	N	O	P
a	p	0 0 0.0 0 0 0.0 1.0	1900.0	g	Первое января 1993 года, 0 часов	2000.0	e	h

```

// Actions
switch (c) {

    case 'r':

        for (int i = 0; i < 30; i++) {

            cout << i + 1 << " step" << "\n";
            cout << "Initial:\n";

            double RA = 15.0 * Rad * Ddd(D: 0, M: i, S: 0.0);
            double Dec = Rad * Ddd(D: 0, M: 0, S: 0.0);
            Vec3D m_R = Vec3D(polar: Polar(Az: RA, Elev: Dec, R: 1.0));

            Pos.m_R = m_R;

            Pos.Print();

            Pos.SetEquinox(T_Equinox: (2000.0 - 2000.0) / 100.0);
            Pos.SetRefSys(Ecliptic);
            Pos.SetOrigin(Heliocentric);
            Pos.SetRefSys(Equator);
            Pos.SetEquinox(T_Equinox: (1900.0 - 2000.0) / 100.0);
            Pos.SetOrigin(Geocentric);
            cout << "After processing:\n";

            abs << accuracy_abs(first: m_R, second: Pos.m_R);
            otn << accuracy_otn(first: m_R, second: Pos.m_R);

```

Рисунок 4 – Исходный код испытаний

Исходные и полученные значения представлены на рис. 5.

1,0,0	1,-2.05216e-10,3.74101e-10
0.99999,0.00436331,0	0.99999,0.00436331,3.741e-10
0.999962,0.00872654,0	0.999962,0.00872654,3.741e-10
0.999914,0.0130896,0	0.999914,0.0130896,3.741e-10
0.999848,0.0174524,0	0.999848,0.0174524,3.741e-10
0.999762,0.0218149,0	0.999762,0.0218149,3.74099e-10
0.999657,0.0261769,0	0.999657,0.0261769,3.74099e-10
0.999534,0.0305385,0	0.999534,0.0305385,3.74099e-10
0.999391,0.0348995,0	0.999391,0.0348995,3.74099e-10
0.999229,0.0392598,0	0.999229,0.0392598,3.74099e-10
0.999048,0.0436194,0	0.999048,0.0436194,3.74098e-10
0.998848,0.0479781,0	0.998848,0.0479781,3.74098e-10
0.99863,0.052336,0	0.99863,0.052336,3.74098e-10
0.998392,0.0566928,0	0.998392,0.0566928,3.74098e-10
0.998135,0.0610485,0	0.998135,0.0610485,3.74098e-10
0.997859,0.0654031,0	0.997859,0.0654031,3.74098e-10
0.997564,0.0697565,0	0.997564,0.0697565,3.74097e-10
0.99725,0.0741085,0	0.99725,0.0741085,3.74097e-10
0.996917,0.0784591,0	0.996917,0.0784591,3.74097e-10
0.996566,0.0828082,0	0.996566,0.0828082,3.74097e-10
0.996195,0.0871557,0	0.996195,0.0871557,3.74096e-10
0.995805,0.0915016,0	0.995805,0.0915016,3.74096e-10
0.995396,0.0958458,0	0.995396,0.0958458,3.74096e-10
0.994969,0.100188,0	0.994969,0.100188,3.74096e-10
0.994522,0.104528,0	0.994522,0.104528,3.74096e-10
0.994056,0.108867,0	0.994056,0.108867,3.74095e-10
0.993572,0.113203,0	0.993572,0.113203,3.74095e-10
0.993068,0.117537,0	0.993068,0.117537,3.74095e-10
0.992546,0.121869,0	0.992546,0.121869,3.74095e-10
0.992005,0.126199,0	0.992005,0.126199,3.74095e-10

Рисунок 5 – Исходные и полученные значения

Для координат x и y разница не видна, так как значения различаются в 10 цифре после запятой.

Погрешность обусловлена ошибками округления вещественных чисел в процессе вычислений с использованием вычислительной техники.

Абсолютная и относительная погрешности представлены на рис. 6 и 7 соответственно.

case		ax
0	1	2.135440e-10
1	2	2.135650e-10
2	3	2.137970e-10
3	4	2.136070e-10
4	5	2.140080e-10
5	6	2.138810e-10
6	7	2.135230e-10
7	8	2.135860e-10
8	9	2.140290e-10
9	10	2.137120e-10
10	11	2.139230e-10
11	12	2.136490e-10
12	13	2.139450e-10
13	14	2.141140e-10
14	15	2.139020e-10
15	16	2.138390e-10
16	17	2.137340e-10
17	18	2.136910e-10
18	19	2.140500e-10
19	20	2.138180e-10
20	21	2.136700e-10
21	22	2.136280e-10
22	23	2.139660e-10
23	24	2.140930e-10
24	25	2.140710e-10
25	26	2.138600e-10
26	27	2.139870e-10
27	28	2.137550e-10
28	29	2.137760e-10
29	30	2.141350e-10

Рисунок 6 – Абсолютная погрешность

case		rx
0	1	2.140870e-10
1	2	2.151480e-10
2	3	2.140800e-10
3	4	2.141160e-10
4	5	2.141270e-10
5	6	2.142050e-10
6	7	2.141100e-10
7	8	2.140900e-10
8	9	2.142770e-10
9	10	2.140810e-10
10	11	2.144700e-10
11	12	2.149680e-10
12	13	2.140830e-10
13	14	2.143190e-10
14	15	2.148050e-10
15	16	2.147290e-10
16	17	2.148840e-10
17	18	2.146580e-10
18	19	2.144160e-10
19	20	2.152440e-10
20	21	2.140960e-10
21	22	2.145910e-10
22	23	2.141750e-10
23	24	2.145290e-10
24	25	2.141010e-10
25	26	2.150560e-10
26	27	2.143650e-10
27	28	2.141490e-10
28	29	2.142390e-10
29	30	2.141350e-10

Рисунок 7 – Относительная погрешность

Графики зависимости абсолютной и относительной погрешностей от выбранного параметра представлены на рис. 8 и 9 соответственно.

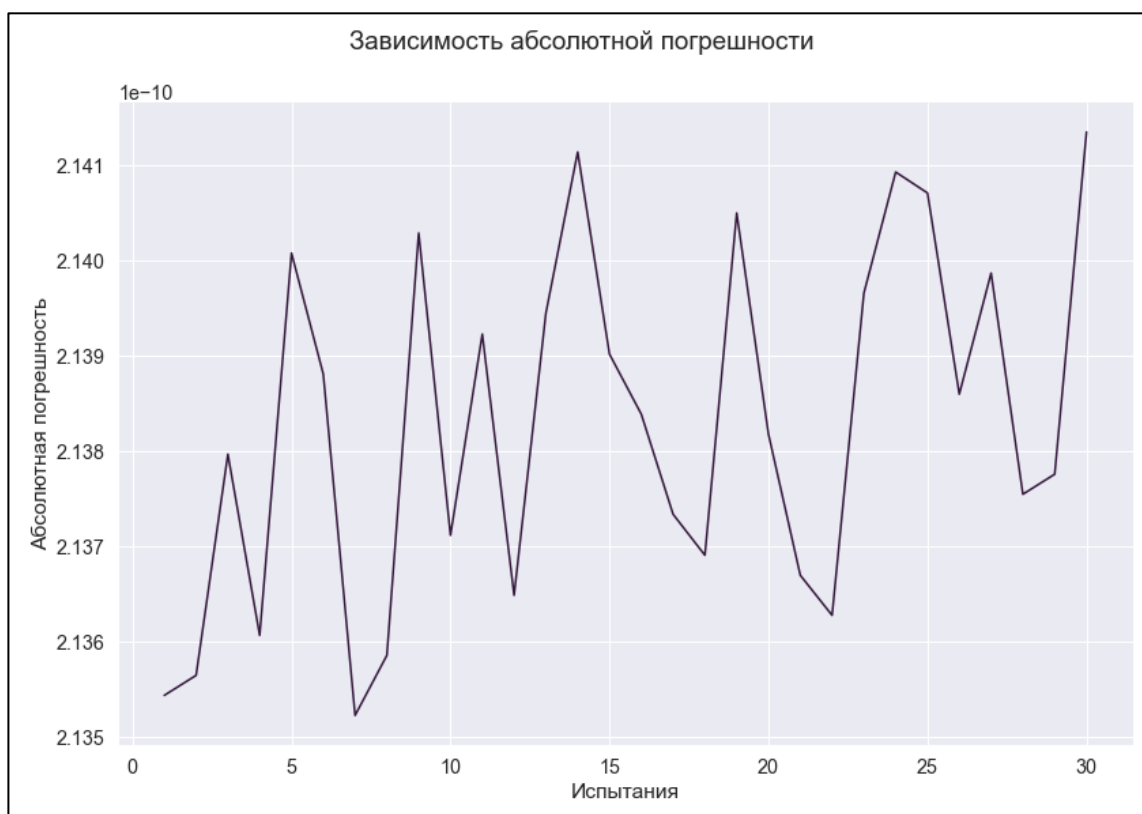


Рисунок 8 – Зависимость абсолютной погрешности

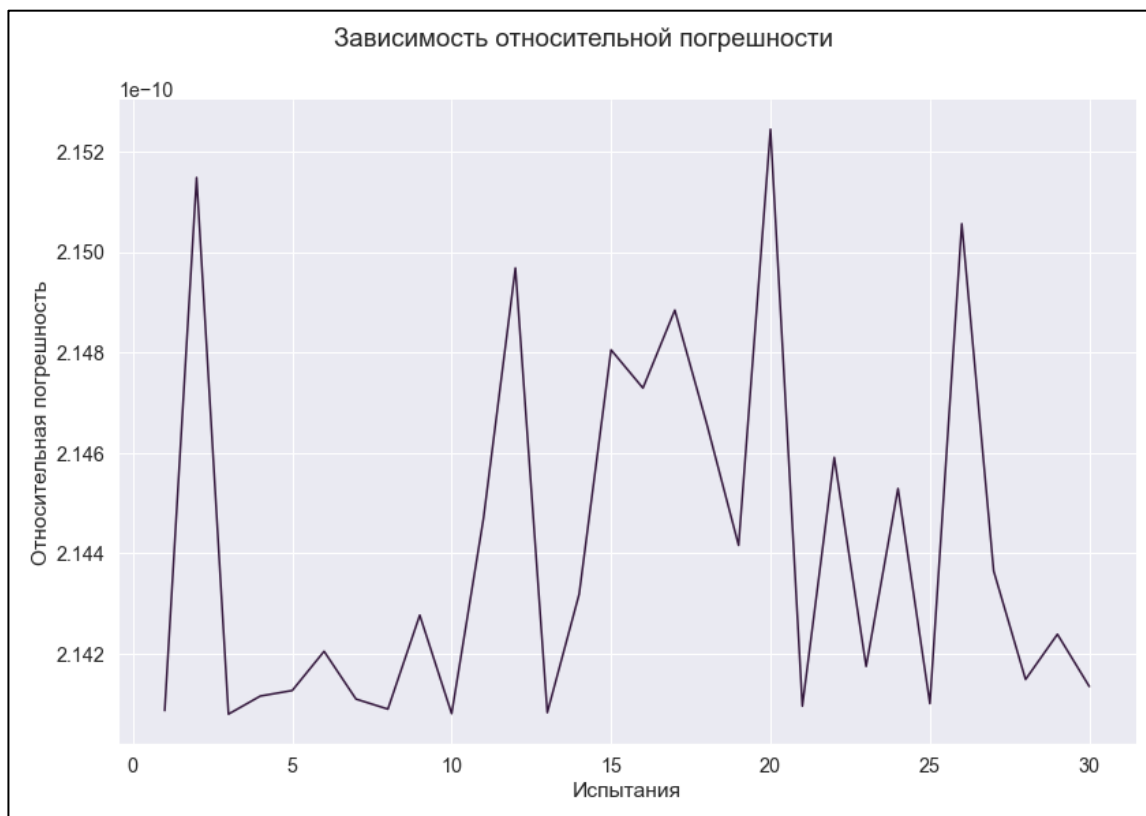


Рисунок 9– Зависимость относительной погрешности

Значения абсолютной погрешности находятся в диапазоне $[2.13523e - 10; 2.14135e - 10]$, а относительной $[2.1408e - 10; 2.15244e - 10]$.

Индивидуальное задание

Задано календарное время X. Вывести значение календарного времени, юлианской даты и модифицированной юлианской даты, соответствующим X.

Модифицированная юлианская дата: $MJD = JD - 2400000.5$.

С помощью программы, представленной в приложении А, были найдены значения юлианской и модифицированной юлианской даты.

Вывод представлен на рис. 10.

```
CD: 1950y 10m 10d 10h 10m 10s
JD: 2433564.9
MJD: 33564.424
```

Рисунок 10 – Календарное время, JD и MJD

Выводы

В ходе данной лабораторной работы была исследована задача преобразования астрономических координат, используя инструментальные средства.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
case 'l':
    cout << "\nCD: 1950y 10m 10d 10h 10m 10s\n";
    mjd = Mjd(1950, 10, 10, 10, 10, 10);
    jd = mjd + 2400000.5;
    cout << "JD: " << jd << "\n";
    cout << "MJD: " << mjd << "\n\n";
    break;

double Mjd ( int Year, int Month, int Day,
             int Hour, int Min, double Sec )
{
    //
    // Variables
    //
    long    MjdMidnight;
    double  FracOfDay;
    int     b;

    if (Month<=2) { Month+=12; --Year;}

    if ( (10000L*Year+100L*Month+Day) <= 15821004L )
        b = -2 + ((Year+4716)/4) - 1179;      // Julian calendar
    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;
}
```