**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 в заголовочных файлах и . Изменения представлены на рис. 1.

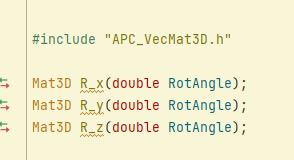


Рисунок 1 – Объявление методов

* Не объявлено пространство имен в .



Рисунок 2 – Пространство имен

* Неверный тип возвращаемого значения функцией

Изменим возвращаемый функцией void тип в файле Coco.cpp

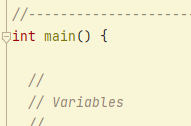
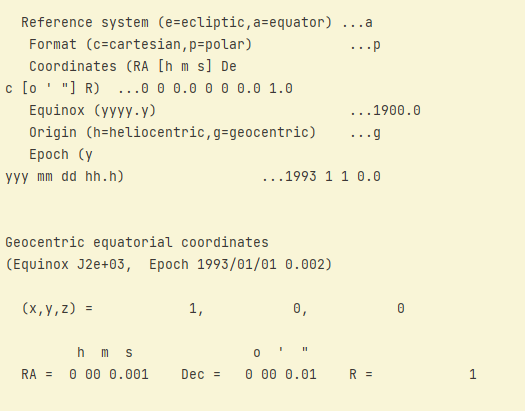


Рисунок 3 – Тип возвращаемого значения

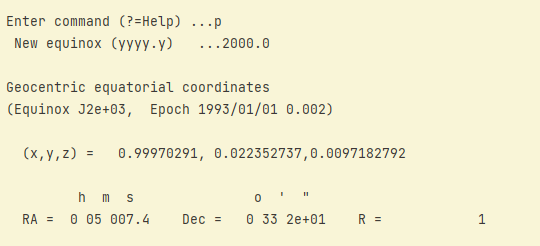
***Циклические преобразования***

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.

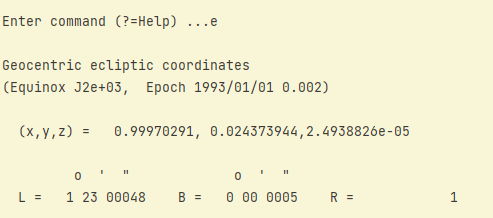
Выбраны экваториальные координаты точки весеннего равноденствия, заданные в полярной форме в эпоху 1900.0, расстояние 1 а.е.



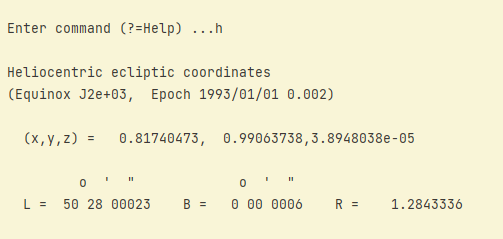
Рассчитаны координаты точки весеннего равноденствия 1900.0, отнесенные к новой эпохе 2000.0:



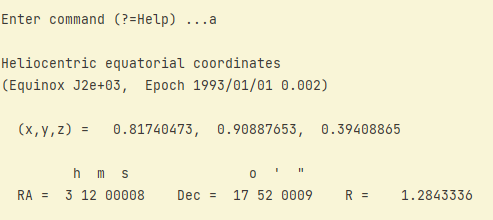
Преобразование экваториальных координат в эклиптические:



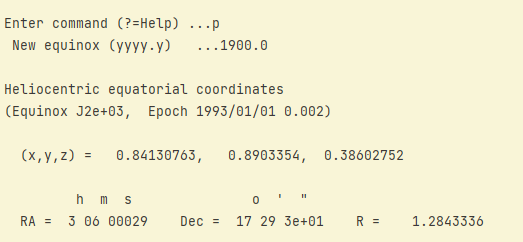
Преобразование координат из геоцентрических в гелиоцентрические на заданную эпоху:



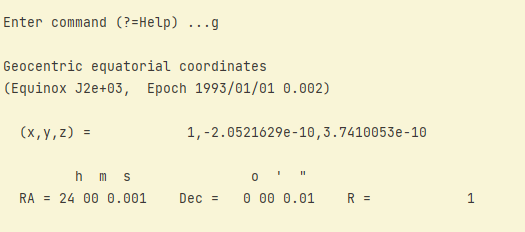
Отмена преобразования в эклиптические координаты:



Пересчет координат к исходной эпохе 1900.0:



Преобразование в геоцентрические координаты:



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 |

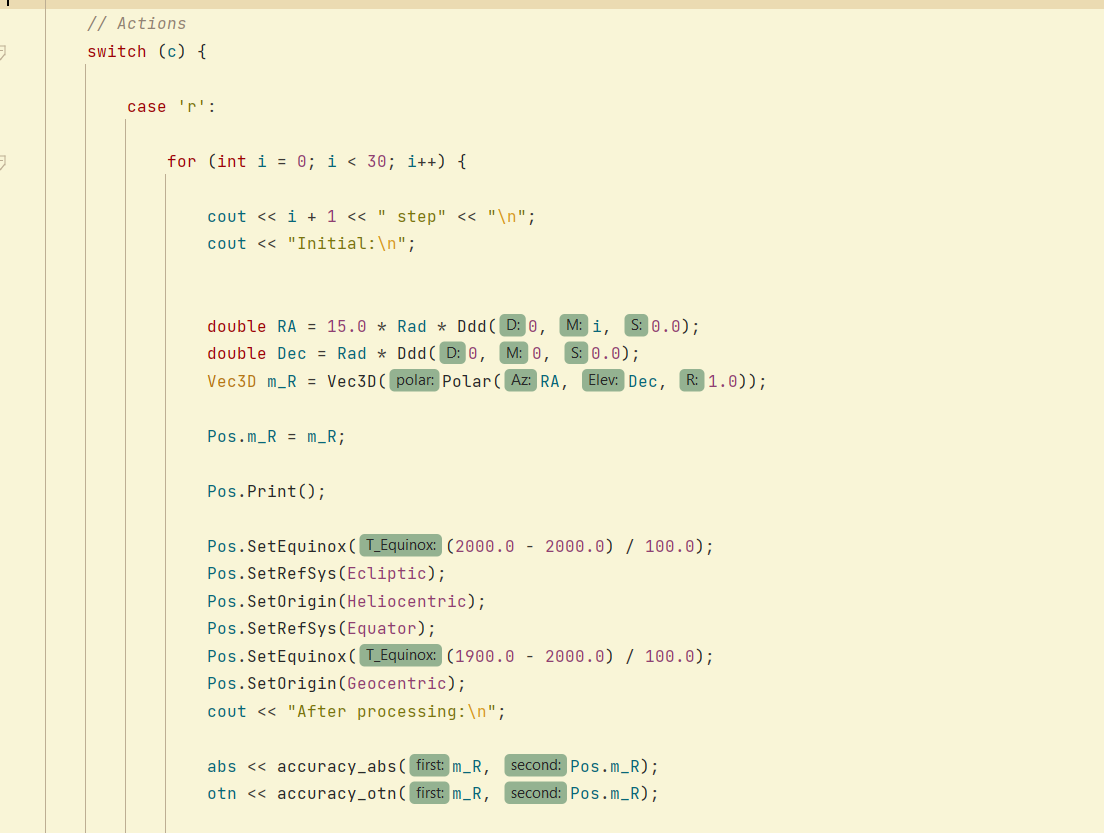


Рисунок 4 – Исходный код испытаний

Исходные и полученные значения представлены на рис. 5.

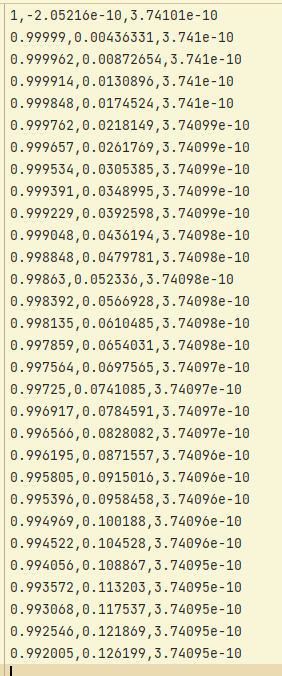
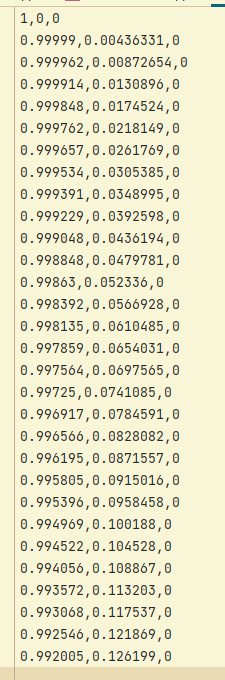


Рисунок 5 – Исходные и полученные значения

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.

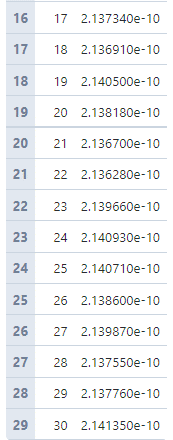
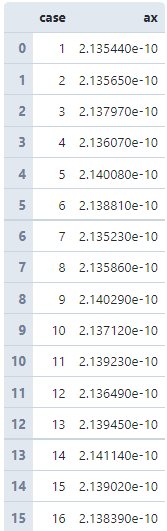


Рисунок 6 – Абсолютная погрешность

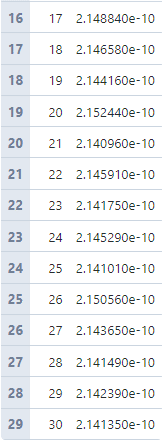
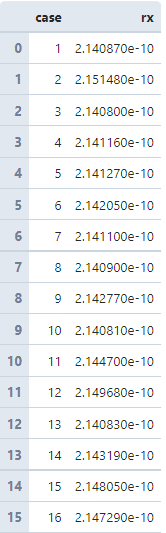


Рисунок 7 – Относительная погрешность

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

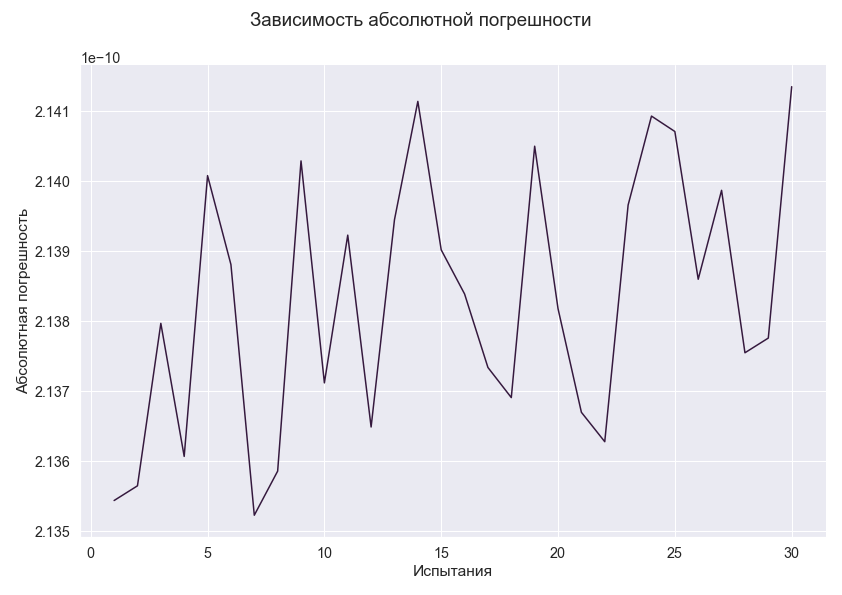


Рисунок 8 – Зависимость абсолютной погрешности

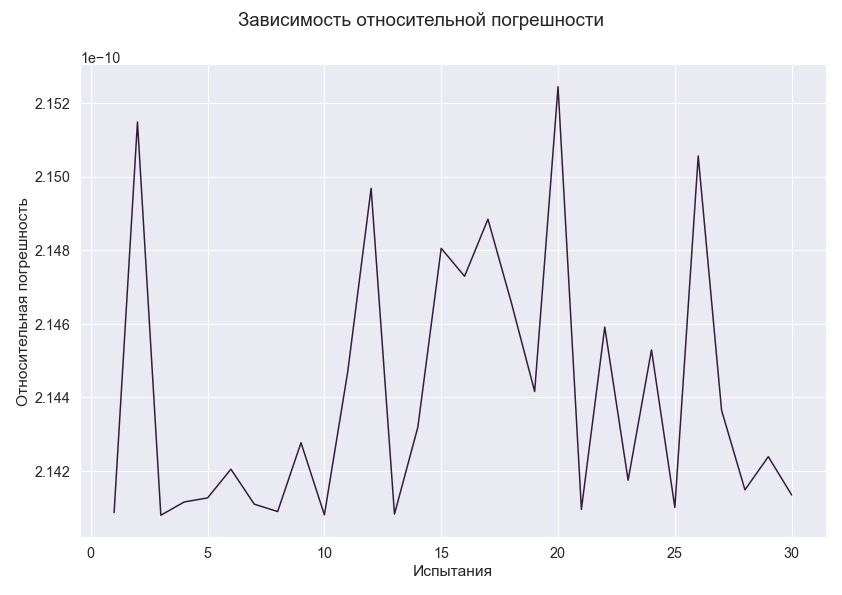


Рисунок 9– Зависимость относительной погрешности

Значения абсолютной погрешности находятся в диапазоне , а относительной .

***Индивидуальное задание***

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

Модифицированная юлианская дата: .

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

Вывод представлен на рис. 10.

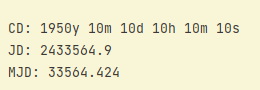


Рисунок 10 – Календарное время, JD и MJD

**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;

}