МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт математики, механики и компьютерных наук им. И.И.Воровича Кафедра математического моделирования

Проект второго курса

Отчет

Направление подготовки Прикладная математика и информатика Направленность программы 01.03.02 Прикладная математика и информатика

|  |  |  |
| --- | --- | --- |
| Обучающийся |  | C.С. Сальников |
| Обучающийся |  | Д.Н. Дурицкий |
| Обучающийся |  | А.М. Меркулов |
| Обучающийся |  | Д.В. Радионов |
| Руководитель |  | О.Г. Пустовалова |

Ростов-на-Дону 2022

Вариант 3

1 Задание NumPy

1.1 Постановка задачи

Сформировать 50 файлов, в каждый из которых записать матрицу 100х100 из случайных целых чисел в диапазоне от -100 до 100. Для матриц сравнить время работы алгоритмов(согласно номеру варианта), реализованных в MATLAB и Python (Numpay).Результаты записать во вновь созданные 50 файлов.Оформить отчет в MicrosoftWord.

Алгоритм - Вектор уникальных значений в массиве.

1.2 Код Matlab

clc

clear

fID = fopen("indreznumpy.txt", 'w'); % Файл со всеми временами выполнения для всех матриц

for i = 1:50

A = randi([-100 100], 100);

fileID = fopen("ind"+string(i)+".txt", 'w'); % Файл с матрицей

fprintf(fileID, '%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d

%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d\n', A);

fclose(fileID);

tic

fileID = fopen("ind"+string(1)+".txt", 'r');

A1 = fscanf(fileID, '%d\n' ,[100 100]);

fclose(fileID);

x = unique(A1);

fid = fopen("numpyrez"+string(i)+".txt", "w"); % Файл с

результатом для одной матрицы

fprintf(fid, '%f\n', x);

fclose(fid);

a = toc

fprintf(fID, string(i)+'. %f\n', a);

end

fclose(fID);

1.3 Код Python

import numpy as np

from datetime import datetime

import time

f = open('indreznumpy\_py.txt', 'w')

for i in range(1, 51):

start\_time = datetime.now()

a = np.genfromtxt('ind' + str(i) + '.txt', delimiter=' ')

res = np.unique(a)

np.savetxt('numpyrez' + str(i) + '\_py.txt', res, fmt = '%11.6f')

f.write(str(i)+ '. ' + str(datetime.now() - start\_time)[6::] + '\n')

f.close()

1.4 Сравнение результатов



2 Задание SymPy

2.1 Постановка задачи

Вычисления выполнять в MATLAB и Python(Sumpay).Вычислить неопределенныйинтеграл.В цикле вычислить определенныеинтегралыдля ста вариантов верхнего и нижнего пределов(пределы и константы(a,b,...)задатьсамостоятельно).Результаты записать втекстовыйфайл.Сравнить результаты вычисленияи время вычисленийвMATLAB и Python. Подынтегральная функция 1/(a\*x+b).

2.2 Код Matlab

clc

clear;

syms x

a = 2

b = 1

f = a \* x + b

int(1 / f, x)

lim1 = randi([1 100], 1, 100);

lim2 = randi([1 100], 1, 100);

fid = fopen("number\_2.txt", "w")

fid2 = fopen("timeNumber2.txt", "w")

tic

for i = 1:100

expr = string(int(1/ f, [lim1(i), lim2(i)])) + newline;

fwrite(fid, expr);

end

t = toc

fwrite(fid2, string(t));

fclose(fid)

fclose(fid2)

2.2 Код Python

import sympy as sp

import random

import time

timing = time.time()

a = int(input("a >> "))

b = int(input("b >> "))

while (a == 0 and b == 0):

a = int(input("a >> "))

b = int(input("b >> "))

lim1 = [random.randint(1, 1000) for i in range(100)]

lim2 = [random.randint(1, 1000) for i in range(100)]

x = sp.Symbol("x")

f = open('number\_3.txt', 'w')

for i in range(100):

if(a != 0):

ans = sp.integrate(1 / (a \* x + b), (x, lim1[i], lim2[i]))

else:

ans = b \* lim2[i] - lim1[i] \* b

f.write(str(ans) + '\n')

f.write(str(time.time() - timing) + " seconds. Programm end")

f.close()

2.3 Сравнение результатов



3 Вывод

Сравнив результаты работы программ, можем сделать вывод, что программы на Python работают быстрее.

4 Литература

1. https://docs.scipy.org/doc/numpy/user/index.html

2. https://docs.scipy.org/doc/numpy/reference/index.html

3. https://www.sympy.org/ru/

4. https://docs.sympy.org/latest/modules/integrals/integrals.html

5. https://docs.sympy.org/latest/modules/integrals/integrals.html#numeric-integrals