ЛАБОРАТОРНАЯ РАБОТА 1. ТЕОРИЯ ПОГРЕШНОСТЕЙ И МАШИННАЯ АРИФМЕТИКА

Евтеева Марина, БПМ-143

Вариант 7

2017

Оглавление

Задание 1.1.7 Дан ряд $\sum_{n=0}^{\infty} a_n$. Найти сумму ряда аналитически. Вычислить значения частичных сумм ряда $S_n = \sum_{n=0}^{N} a_n$ и найти величину погрешности при значениях $N = 10, 10^2, 10^3, 10^4, 10^5$.
Вариант 7:

$$a_n = \frac{24}{n^2 + 8n + 15}$$

Решение:

1. Находим сумму ряда аналитически, как предел частичных сумм: $S_n = \sum_{n=0}^{N} a_n = \sum_{n=0}^{N} \frac{24}{n^2+8n+15} = \sum_{n=0}^{N} \frac{24}{(n+3)(n+5)} = 12 * \sum_{n=0}^{N} \frac{1}{n+3} - \frac{1}{n+5} = \frac{1}{3} - \frac{1}{5} + \frac{1}{4} - \frac{1}{6} + \frac{1}{5} \cdots + \frac{1}{n+3} - \frac{1}{n+5})$
$S = \lim_{N \to \infty} S_N = 7$
Ответ: $\sum_{n=0}^{\infty} \frac{24}{n^2+8n+15} = 7$

2. Частичные суммы вычисляются с помощью кода MATLAB

```
syms n
for i = 1:5
    double(symsum(24/(n^2 + 8*n + 15), 0, 10^i))
end
```

Получившиеся значения:
$S(10^1) = 5.342857142857143$
$S(10^2) = 6.770329670329670$
$S(10^3) = 6.976107510257478$
$S(10^4) = 6.997601079508227$
$S(10^5) = 6.999760010799508$

3. Соответствующие величины абсолютной погрешности:
$d(10^1) = 1.6571$
$d(10^2) = 0.22967$
$d(10^3) = 0.02389$
$d(10^4) = 0.002399$
$d(10^5) = 0.000239989$
Количество верных цифр:
$M_1 = 1$
$M_2 = 2$
$M_3 = 3$
$M_4 = 4$
$M_5 = 5$
Гистограмма, показывающая количество верных цифр рис.1

Задание 1.8 Составить программу, моделирующую вычисления на ЭВМ с ограниченной разрядностью m. Решить задачу 1.1 для случая $N = 10000$, используя эту программу. Составить график зависимости погрешности от количества разрядов $m = 4, 5, \cdots, 8$. **Решение:** Код программы на MATLAB

```
d = [];
p = [];
for m = 4:8
    sn = 0;
    s = 7 + 1*10^(-m);
    for n = 0:10000
        sn = sn + round(24/(n^2 + 8*n + 15)*10^m)/10^m;
    end
    d = [d;round(abs(sn - s)*10^m)/10^m];
    p = [p; round(d(m-3)/s*10^m)/10^m]
end
plot([4, 5, 6, 7, 8], p, 'ro--')
```

Получившаяся зависимость погрешности от количества разрядов представлена на графике рис.2

Задание 1.4.2 Найти ранг заданной матрицы A. Затем внести погрешность в 0.1% а) в элемент a_{11}; b) во все элементы матрицы и снова найти ранг. Объяснить полученные результаты.
Вариант 7

$$\begin{matrix} 0.6 & 4.5 & 0.3 & 3 \\ -2.4 & -12 & 0.9 & -7 \\ 1.2 & 9 & 0.6 & 6 \\ -1.2 & 3 & 3.6 & 4 \end{matrix}$$

Решение:
Код MATLAB, реализующий решение данной задачи

```
A = [0.6 4.5 0.3 3;
-2.4 -12 0.9 -7;
 1.2 9 0.6 6;
-1.2 3 3.6 4];
rank(A)
B = A;
B(1, 1) = A(1,1)+A(1,1)*0.1/100;
rank(B)
C = A;
for i = 1:4
    for j = 1:4
        C(i, j) = A(i, j) + A(i, j)*0.1/100;
    end
end
rank(C)
```

Ранг матрицы А при этом получился = 2, ранг матрицы, с внесенной в a_{11} погрешностью в 0.1% получился равен 3, но если вносить погрешность в 0.1% в каждый элемент матрицы, то ее ранг снова становится равен 2 Это происходит из-за того, что при внесении только в один элемент погрешности мы нарушаем линейную независимость векторов (погрешность в 0.1% довольно велика), а внося во всю матрицу - линейная независимость сохраняется

Задание 1.7 Вычислить значения машинного нуля, машинной бесконечности, машинного эпсилон в режимах одинарной и двойной точности на алгоритмическом языке. Сравнить результаты с результатами задачи 1.6. УКАЗАНИЕ: при использовании языка Python, вещественные числа ординарной точности можно получить используя библиотеку NumPy, напр. np.float32(1) и т.д.

Решение:

Текст программы на языке Pyhton

```
import numpy as np
import math

np.seterr(over='raise', under='raise')
N = 1
x = np.float64(1)
while x != math.inf:
    try:
        x *= np.float64(2)
        N += 1
    except:
        break
print(2.0**(N-1))
n = 1
x1 = np.float64(1)
while not (x1 == np.float64(0)):
    try:
        x1 /= np.float64(2)
        n += 1
    except:
        break
print(2**(-n+1))
k = 0
x2 = np.float64(1)
y2 = np.float64(2)
while y2 != np.float64(1):
    x2 /= np.float64(2)
    y2 = np.float64(1) + x2
    k += 1
print(2**(-k+1))
N = 1
x = np.float32(1)
while x != math.inf:
    try:
        x *= np.float32(2)
        N += 1
    except:
        break
print(2.0**(N-1))
```

```
n = 1
x1 = np.float32(1)
while not (x1 == np.float32(0)):
    try:
        x1 /= np.float32(2)
        n += 1
    except:
        break
print(2**(-n+1))
k = 0
x2 = np.float32(1)
y2 = np.float32(2)
while y2 != np.float32(1):
    x2 /= np.float32(2)
    y2 = np.float32(1) + x2
    k += 1
print(2**(-k+1))
```

Получившиеся значения:
Машинная бесконечность:
При одинарной точности $X_\infty = 1.7014118346046923e + 38$
При двойной точности $X_\infty = 8.98846567431158e + 307$
Машинный нуль:
При одинарной точности $X_0 = 1.401298464324817e - 45$
При двойной точности $X_0 = 5e - 324$
Машинное эпсилон:
При одинарной точности $\epsilon = 1.1920928955078125e - 07$
При двойной точности $\epsilon = 2.220446049250313e - 16$

Задание 1.6 Для пакета MATHCAD найти значения машинного нуля, машинной бесконечности, машинного эпсилон
Решение:
Текст программы на MATLAB (значения находились для него)

```
k = 0;
while 1 + 2.0 ^ (-k) ~= 1
    k = k + 1;
end
eps = 2 ^ (-k + 1)

n = 0;
while 2.0 ^ (-n) ~= 0
    n = n + 1;
end
zero = 2 ^ (-n + 1)

n = 0;
while 2.0^n ~= inf
```

```
    n = n+1;
end
 infin = 2.0 ^ (n - 1)
```

Получившиеся значения:
Машинная бесконечность:
$X_\infty = 8.988465674311580e + 307$
Машинный нуль:
$X_0 = 4.940656458412465e - 324$
Машинное эпсилон:
$\epsilon = 2.220446049250313e - 16$

Задание 1.9.1 Для матрицы A решить вопрос о существовании обратной матрицы в следующих случаях: 1) элементы матрицы заданы точно; 2) элементы матрицы заданы приближенно с относительной погрешностью a) $\delta = \alpha\%$ и b) $\delta = \beta\%$. Найти относительную погрешность результата
Вариант 7

$$A = \begin{pmatrix} 31 & 27 & 22 \\ 32.2 & 28.2 & 24 \\ 36 & 32 & 27 \end{pmatrix}$$

$\alpha = 0.1$
$\beta = 0.4$
Решение:
Обратная матрица существует тогда и только тогда, когда матрица невырожденная, то есть ее определитель не равен 0. Когда элементы матрицы заданы точно найти определитель несложно. Когда элементы матрицы заданы с относительной погрешностью δ, то мы не знаем конкретных значений, а знаем лишь только, что он принимает значения из отрезка. А определитель это непрерывная и дефференциируемая функция, значит применив теорему Вейерштрассе мы можем понять, есть ли нуль в нашем множестве значений функции определителя. Значит нам нужно найти все возможные значения определителя в точках, координаты которых имеют вид $a_{ij}(1 \pm \delta)$, выбрать из них наименьший и наибольший и посмотреть на их знаки.
Код программы на MATLAB, реализующий поиск минимального и максимального определителя (и для α, и для β):

```
A = [31 27 22;
32.2 28.2 24;
36 32 27];
alpha = 0.1;
beta = 0.4;
det(A)
flag = zeros(2, 9);
flag(1, :) = 1;
dets1 = [];
dets2 = [];
for i = 1:512
    B = A;
    C = A;
```

```
    for j = 1:3
        for k = 1:3
            B(j, k) = (1 + flag(1, 3*(j-1) + k)*alpha) * B(j, k);
            C(j, k) = (1 + flag(1, 3*(j-1) + k)*beta) * C(j, k);
        end
    end
    dets1 = [dets1; det(B)];
    dets2 = [dets2; det(C)];
    flag(2, :) = flag(2, :) + 1;
    for j = 1:9
        if flag(2, j) == 2^(j-1)
            flag(2, j) = 0;
            flag(1, j) = -flag(1, j);
        end
    end
%     flag(1, :)
end
min(dets1)
max(dets1)
min(dets2)
max(dets2)
```

Получившиеся значения:
Определитель матрицы, заданной точно:
-16.000000000000060
Минимальный и максимальный определители при $\delta = \alpha$:
$-3.419235600000005e + 03$
$3.179206800000000e + 03$
Минимальный и максимальный определители при $\delta = \beta$:
$-5.271565440000000e + 04$
$5.180757119999998e + 04$
В обоих случаях минимальный и максимальный определители разных знаков, что создает неопределенность: где-то функция определителя обращается в нуль, значит для некоторых матриц мы не можем найти обратные.
Относительные погрешности для минимальных и максимальных значений определителя при обеих δ:
$min, \delta = \alpha,\ 2.127022249999995e + 02$
$max, \delta = \alpha,\ 1.997004249999992e + 02$
$min, \delta = \beta,\ 3.293728399999988e + 03$
$max, \delta = \beta,\ 3.238973199999987e + 03$

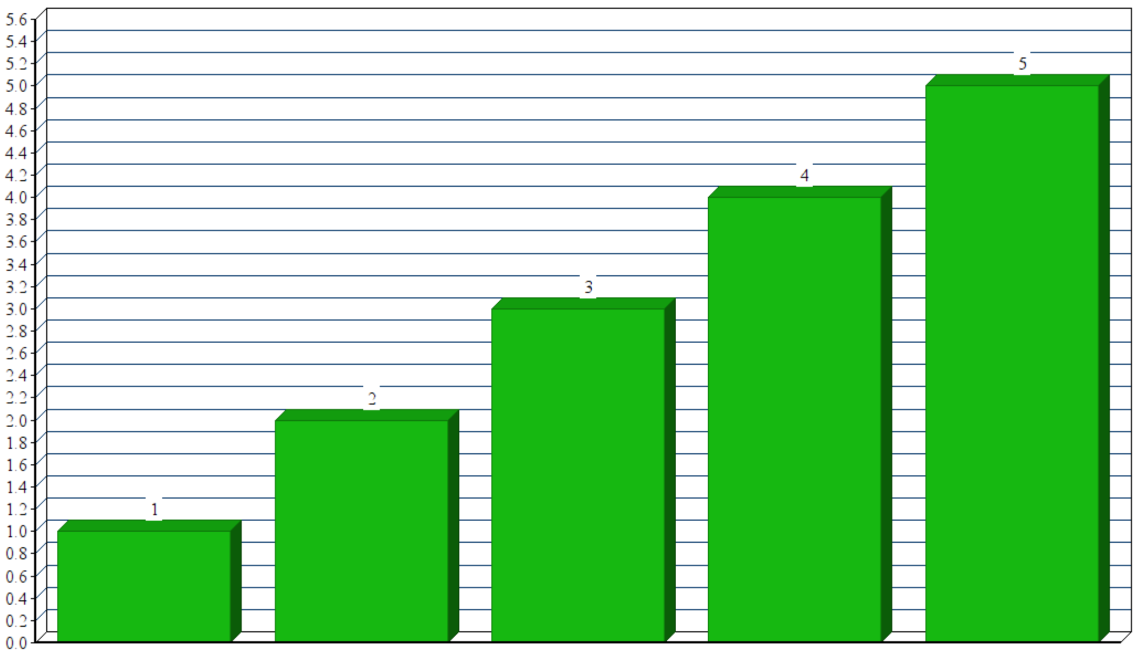

Рис. 1: Гистограмма, показывающая количество верных цифр

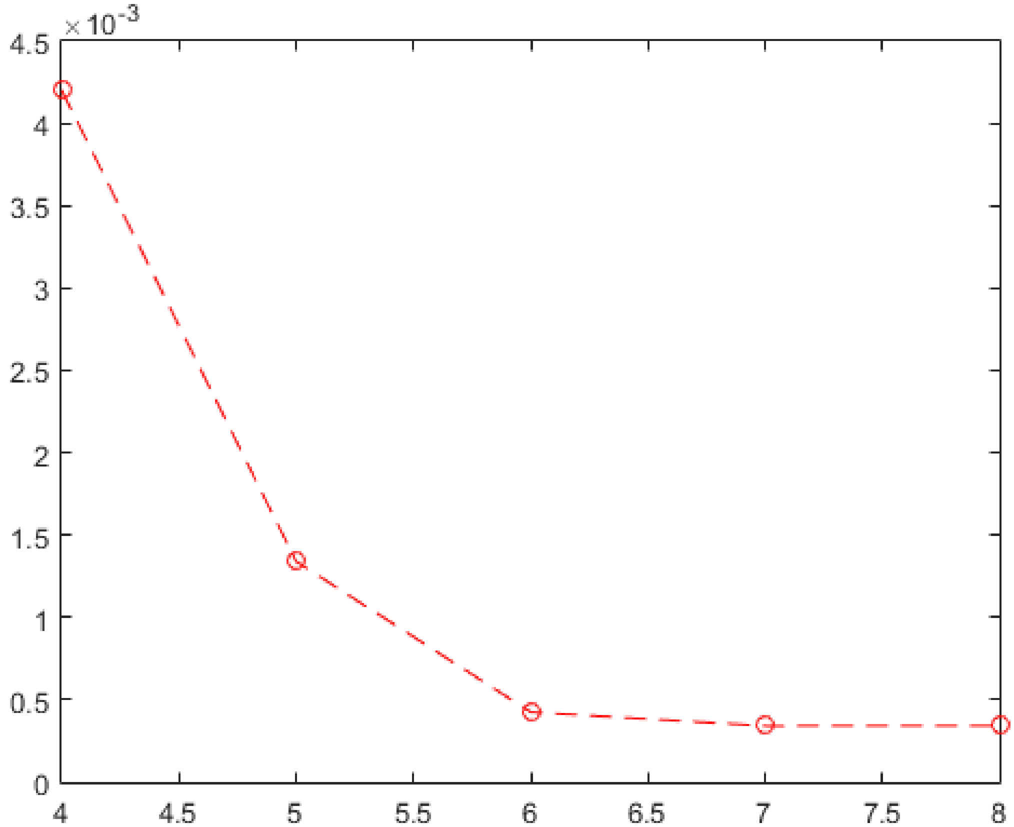

Рис. 2: Зависимость погрешности от количества разрядов