**Выполните в Matlab команду p=poly(1:20), которая позволяет получить коэффициенты полинома, корнями которого является аргумент функции. Наберите roots(p) и убедитесь, что корни полинома найдены верно. Измените значение, например, второго коэффициента на малую величину 10^−7 (составляет ≈ 5 · 10−8% от числа) и снова выполните эту команду - половина корней стала комплексными числами!**

>> p=poly(1:20)

p =

1.0e+19 \*

Columns 1 through 9

0.0000 -0.0000 0.0000 -0.0000 0.0000 -0.0000 0.0000 -0.0000 0.0000

Columns 10 through 18

-0.0000 0.0001 -0.0010 0.0063 -0.0311 0.1207 -0.3600 0.8038 -1.2871

Columns 19 through 21

1.3804 -0.8753 0.2433

>> roots(p)

ans =

20.0000

18.9994

18.0033

16.9882

16.0262

14.9564

14.0507

12.9560

12.0300

10.9855

10.0054

8.9986

8.0002

7.0000

6.0000

5.0000

4.0000

3.0000

2.0000

1.0000

>> p(2)=p(2)-10^(-7);

>> roots(p)

ans =

21.0320 + 0.0000i

19.6348 + 2.1403i

19.6348 - 2.1403i

16.7469 + 3.0569i

16.7469 - 3.0569i

13.9344 + 2.7256i

13.9344 - 2.7256i

11.7100 + 1.8105i

11.7100 - 1.8105i

10.0141 + 0.7629i

10.0141 - 0.7629i

8.8769 + 0.0000i

8.0113 + 0.0000i

6.9996 + 0.0000i

6.0000 + 0.0000i

5.0000 + 0.0000i

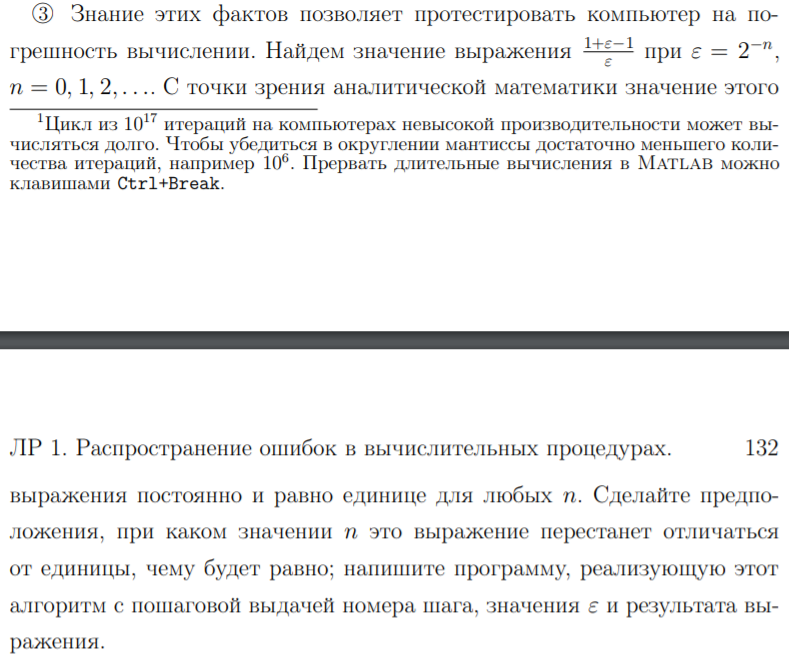
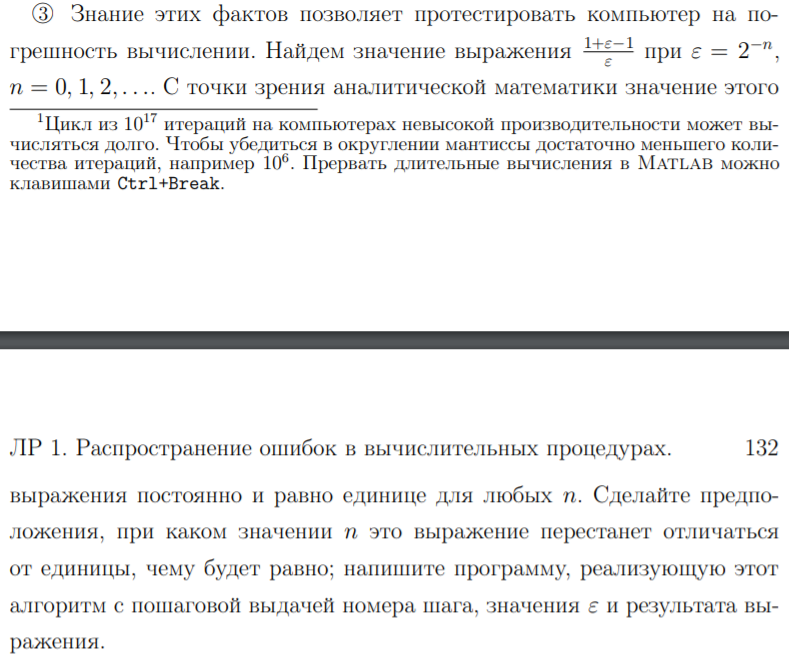
4.0000 + 0.0000i

3.0000 + 0.0000i

2.0000 + 0.0000i

1.0000 + 0.0000i

**Исходная задача оказалась неустойчивой к входным данным (их малое изменение ведет к сильному изменению решения), в результате чего получился абсурдный результат.**



i=0;

eps=2^-i;

while (((1+eps-1)/eps)==1)

i=i+1;

eps=2^-i;

a=(1+eps-1)/eps;

end

i

eps

a

>> ChM1v1

i =

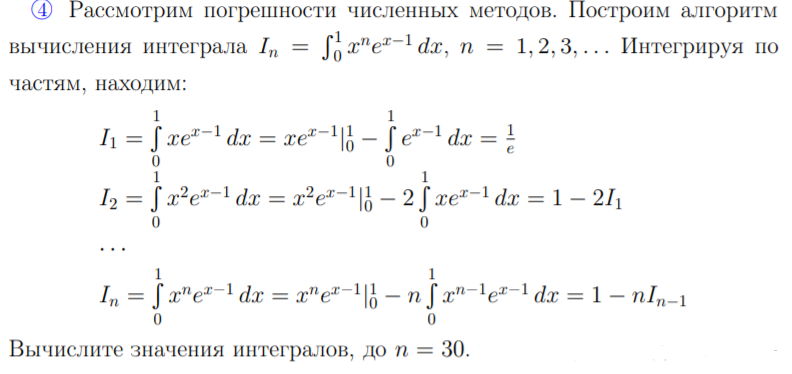
53

eps =

1.110223024625157e-16

a =

0



I=1/exp(1);

for i=2:1:30

I=1-i\*I

end

I2= 2.642411176571154e-01

I3 = 2.072766470286537e-01

I4 = 1.708934118853853e-01

I5 = 1.455329405730734e-01

I6 = 1.268023565615595e-01

I7= 1.123835040690837e-01

I8= 1.009319674473304e-01

I9 = 9.161229297402684e-02

I10= 8.387707025973157e-02

I11 = 7.735222714295276e-02

I 12= 7.177327428456692e-02

I13 = 6.694743430062999e-02

I14 = 6.273591979118009e-02

I15 = 5.896120313229858e-02

I16 = 5.662074988322274e-02

I17 = 3.744725198521337e-02

I18 = 3.259494642661593e-01

I19 = -5.193039821057027e+00

I20 = 1.048607964211405e+02

I21 = -2.201076724843952e+03

I22 = 4.842468794656693e+04

I23 = -1.113766822771040e+06

I24 = 2.673040474650495e+07

I25 = -6.682601176626236e+08

I26 = 1.737476306022821e+10

I27 = -4.691186026251618e+11

I28 = 1.313532087350553e+13

I29 = -3.809243053316594e+14

I30 = 1.142772915994978e+16

**Можно заметить, что до 17 номера значение интеграла уменьшается, а с 18**

**оно начинает резко возрастать.**

**Это происходит из-за того, что изначально мы используем частное от единицы и**

**экспоненты, что само по себе конечной дробью не является, поэтому**

**MATLAB округляет значение каждого интеграла в меньшую сторону (до 16**

**разрядов мантиссы).**

**Получается, что 17-й интеграл выдаёт число, которое в корне меняет конечное**

**значение 30-го интеграла.**

function ChM1v2(x)

u=x;

S=x;

k=0;

while (abs(u)>10^-17)

u=u\*(-x^2)/(2\*k+2)/(2\*k+3);

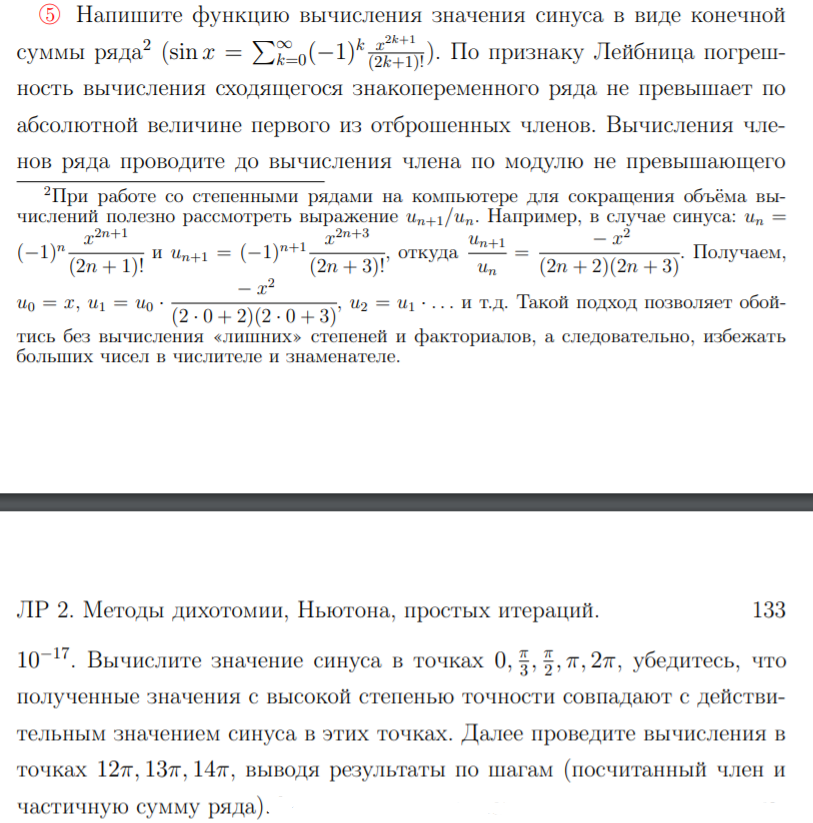
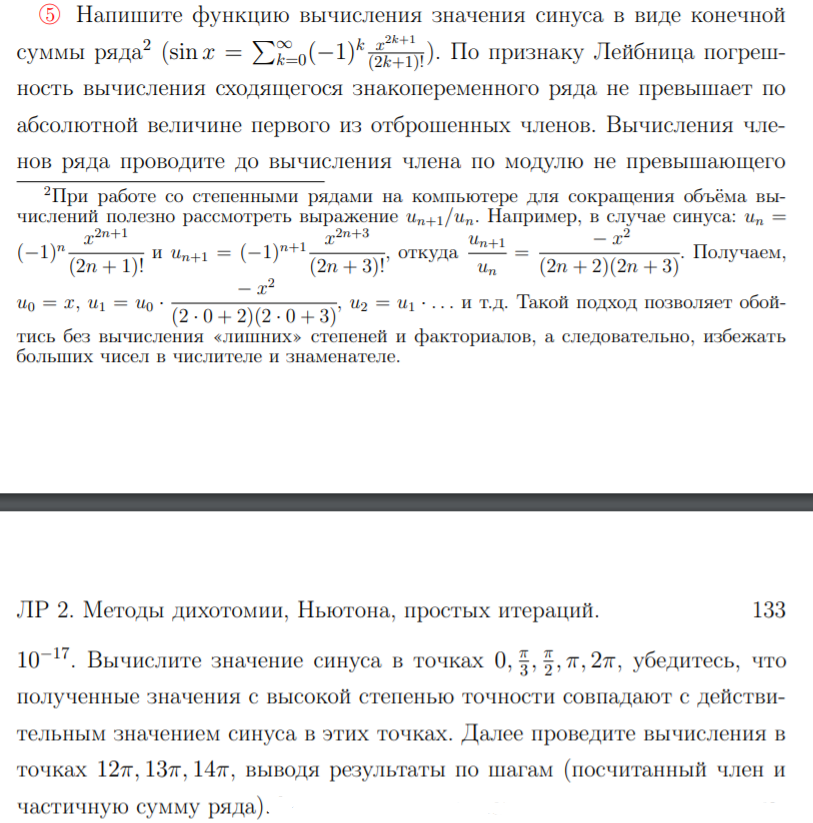
S=S+u;

k=k+1;

end

S

end

function [S,u]=ChM1v2(x)

u=[x];

S=x;

k=0;

while (abs(u)>10^-17)

u1=u(k+1).\*(-x^2)./(2.\*k+2)./(2.\*k+3);

u=[u u1];

S=S+u(k+2);

k=k+1;

end

end

>> ChM1v2(0)

S =

0

>> sin(0)

ans =

0

>> ChM1v2(pi/3)

S =

0.8660

>> sin(pi/3)

ans =

0.8660

>> ChM1v2(pi/2)

S =

1.0000

>> sin(pi/2)

ans =

1

>> sin(pi)

ans =

1.2246e-16

>> ChM1v2(pi)

S =

2.4709e-16

>> ChM1v2(2\*pi)

S =

-8.9503e-16

>> sin(2\*pi)

ans =

-2.4493e-16