Задача 2

1. #include<string>
3. #include <iostream>
4. #include <math.h>
5. using namespace std;
6. double a, b, y, A;
7. void Grad(double x10, double x20, double x30, double e);
8. void print(double k, double x1, double x2, double x3, double f, double pr1, double pr2, double pr3, double ak)
9. {
10. cout << k;
11. printf(" %.7lf", x1);
12. printf(" %.7lf", x2);
13. printf(" %.7lf", x3);
14. printf(" %.7lf", f);
15. printf(" %.10lf", pr1);
16. printf(" %.10lf", pr2);
17. printf(" %.10lf", pr3);
18. printf(" %.10lf", ak);
19. cout << endl;
20. }
21. double func(double x1, double x2, double x3)
22. {
23. double f;
24. f = pow((x1 + a + x2), 4) + pow((x1 + b + x3), 2) + pow((x2 + y + x3), 4);
25. return f;
26. }
27. double dfdx1(double x1, double x2, double x3)
28. {
29. double pr1;
30. pr1 = 4 \* pow((x1 + a + x2), 3) + 2 \* (x1 + b + x3);
31. return pr1;
32. }
33. double dfdx2(double x1, double x2, double x3)
34. {
35. double pr2;
36. pr2 = 4 \* pow((x1 + a + x2), 3) + pow((x2 + y + x3), 3);
37. return pr2;
38. }
39. double dfdx3(double x1, double x2, double x3)
40. {
41. double pr3;
42. pr3 = 2 \* (x1 + b + x3) + 4 \* pow((x2 + y + x3), 3);
43. return pr3;
44. }
45. double PH(double x1, double x2, double x3, double pr1, double pr2, double pr3, double A)
46. {
47. double Ph;
48. Ph = pow((x1 + a + x2 - 2 \* A\*(pr1 + pr2)), 4) + pow((x1 + b + x3 - 2 \* A\*(pr1 + pr3)), 2) + pow((x2 + y + x3 - 2 \* A\*(pr2 + pr3)), 4);
49. return Ph;
50. }
51. double pr1PH(double x1, double x2, double x3, double pr1, double pr2, double pr3, double A)
52. {
53. double pr1Ph;
54. pr1Ph = -8 \* (pr1 + pr2)\*pow((x1 + a + x2 - 2 \* A\*(pr1 + pr2)), 3) - 4 \* (pr1 + pr3)\*(x1 + b + x3 - 2 \* A\*(pr1 + pr3)) - 8 \* (pr2 + pr3)\*pow((x2 + y + x3 - 2 \* A\*(pr2 + pr3)), 3);
55. return pr1Ph;
56. }
57. double pr2PH(double x1, double x2, double x3, double pr1, double pr2, double pr3, double A)
58. {
59. double pr2Ph;
60. pr2Ph = 48 \* pow((pr1 + pr2), 2)\*pow((x1 + a + x2 - 2 \* A\*(pr1 + pr2)), 2) + 8 \* pow((pr1 + pr3), 2) + 48 \* pow((pr2 + pr3), 2)\*pow((x2 + y + x3 - 2 \* A\*(pr2 + pr3)), 2);
61. return pr2Ph;
62. }
63. void Niuton(double x1, double x2, double x3, double pr1, double pr2, double pr3)
64. {
65. int n = 0;
66. double eps, Amin, Amax;
67. double Ph, p\_1, p\_2;
68. eps = 0.001;
69. Amin = 0.001;
70. Amax = 1;
71. A = Amin;
72. Ph = PH(x1, x2, x3, pr1, pr2, pr3, A);
73. p\_1 = pr1PH(x1, x2, x3, pr1, pr2, pr3, A);
74. p\_2 = pr2PH(x1, x2, x3, pr1, pr2, pr3, A);
75. while (fabs(Ph)>eps)
76. {
77. A = A - (p\_1 / p\_2);
78. if (A>Amax)
79. {
80. n++;
81. A = Amax;
82. Ph = PH(x1, x2, x3, pr1, pr2, pr3, A);
83. }
84. if (A>Amax)
85. {
86. n++;
87. A = Amax;
88. Ph = PH(x1, x2, x3, pr1, pr2, pr3, A);
89. }
90. else
91. {
92. n++;
93. A = Amin;
94. Ph = PH(x1, x2, x3, pr1, pr2, pr3, A);
95. }
96. p\_1 = pr1PH(x1, x2, x3, pr1, pr2, pr3, A);
97. p\_2 = pr2PH(x1, x2, x3, pr1, pr2, pr3, A);
98. if (n>1)
99. break;
100. }
101. }
102. int main(int argc, const char \* argv[]) {
103. double x10, x20, x30, e;
105. a = 51;
106. b = 45;
107. y = 88;
108. x10 = -6;
109. x20 = -45;
110. x30 = -43;
111. e = 0.001;
112. int k, i;
113. double x1[100000],A, Ph[10000], x2[100000], x3[100000], pr1[100000], pr2[100000], pr3[100000], ak[100000], f[100000];
114. k = 0;
115. A = 0.001;
116. x1[k] = x10;
117. x2[k] = x20;
118. x3[k] = x30;
119. pr1[k] = dfdx1(x1[k], x2[k], x3[k]);
120. pr2[k] = dfdx2(x1[k], x2[k], x3[k]);
121. pr3[k] = dfdx3(x1[k], x2[k], x3[k]);
122. f[k] = func(x1[k], x2[k], x3[k]);
123. do
124. {
125. Niuton(x1[k], x2[k], x3[k], pr1[k], pr2[k], pr3[k]);
126. ak[k] = A;
127. x1[k + 1] = x1[k] - ak[k] \* pr1[k];
128. x2[k + 1] = x2[k] - ak[k] \* pr2[k];
129. x3[k + 1] = x3[k] - ak[k] \* pr3[k];
130. f[k + 1] = func(x1[k + 1], x2[k + 1], x3[k + 1]);
131. k++;
132. ak[k] = ak[k - 1];
133. pr1[k] = dfdx1(x1[k], x2[k], x3[k]);
134. pr2[k] = dfdx2(x1[k], x2[k], x3[k]);
135. pr3[k] = dfdx3(x1[k], x2[k], x3[k]);
136. }
137. while ((fabs(pr1[k]>e)) || (fabs(pr2[k])>e) || (fabs(pr3[k])>e));
138. cout << "Таблица значений:" << endl;
139. cout << "k x1(k) x2(k) x3(k) f(k) df/dx1 df2/dx2 ak" << endl;
140. cout << "Первые 10 итераций" << endl;
141. for (i = 0; i<10; i++)
142. print(i, x1[i], x2[i], x3[i], f[i], pr1[i], pr2[i], pr3[i], ak[i]);
143. cout << "Последние 10 итераций" << endl;
144. for (i = k - 9; i <= k; i++)
145. print(i, x1[i], x2[i], x3[i], f[i], pr1[i], pr2[i], pr3[i], ak[i]);
146. cin.get();
147. return 0;
149. }

**Таблица значений:**

**k x1(k) x2(k) x3(k) f(k) df/dx1 df2/dx2 ak e**

**Первые 10 итераций:**

**0 -6.0000000 -45.0000000 -43.0000000 16.0000000 -8.0000000000 0.0000000000 -8.0000000000 0.0010000000**

**1 -5.9920000 -45.0000000 -42.9920000 15.8722560 -7.9679979520 0.0000025600 -7.9679979520 0.0010000000**

**2 -5.9840320 -45.0000000 -42.9840320 15.7455321 -7.9361117223 0.0000203573 -7.9361117223 0.0010000000**

**3 -5.9760959 -45.0000000 -42.9760959 15.6198205 -7.9043289256 0.0000682946 -7.9043289256 0.0010000000**

**4 -5.9681916 -45.0000001 -42.9681916 15.4951141 -7.8726375146 0.0001609138 -7.8726375146 0.0010000000**

**5 -5.9603189 -45.0000003 -42.9603189 15.3714061 -7.8410257750 0.0003124007 -7.8410257750 0.0010000000**

**6 -5.9524779 -45.0000006 -42.9524779 15.2486900 -7.8094823216 0.0005365886 -7.8094823216 0.0010000000**

**7 -5.9446684 -45.0000011 -42.9446684 15.1269597 -7.7779960944 0.0008469610 -7.7779960944 0.0010000000**

**8 -5.9368904 -45.0000019 -42.9368904 15.0062097 -7.7465563557 0.0012566538 -7.7465563557 0.0010000000**

**9 -5.9291439 -45.0000032 -42.9291439 14.8864346 -7.7151526874 0.0017784575 -7.7151526874 0.0010000000**

**Последние 10 итераций:**

**31562 -4.0002027 -46.9413087 -41.0002027 0.0000236 -0.0000103810 0.0010004255 -0.0000103810 0.0010000000**

**31563 -4.0002027 -46.9413097 -41.0002027 0.0000236 -0.0000103801 0.0010003747 -0.0000103801 0.0010000000**

**31564 -4.0002027 -46.9413107 -41.0002027 0.0000236 -0.0000103793 0.0010003239 -0.0000103793 0.0010000000**

**31565 -4.0002026 -46.9413117 -41.0002026 0.0000236 -0.0000103784 0.0010002731 -0.0000103784 0.0010000000**

**31566 -4.0002026 -46.9413127 -41.0002026 0.0000236 -0.0000103775 0.0010002224 -0.0000103775 0.0010000000**

**31567 -4.0002026 -46.9413137 -41.0002026 0.0000236 -0.0000103766 0.0010001716 -0.0000103766 0.0010000000**

**31568 -4.0002026 -46.9413147 -41.0002026 0.0000236 -0.0000103757 0.0010001208 -0.0000103757 0.0010000000**

**31569 -4.0002026 -46.9413157 -41.0002026 0.0000236 -0.0000103749 0.0010000700 -0.0000103749 0.0010000000**

**31570 -4.0002026 -46.9413167 -41.0002026 0.0000236 -0.0000103740 0.0010000192 -0.0000103740 0.0010000000**

**31571 -4.0002026 -46.9413177 -41.0002026 0.0000236 -0.0000103731 0.0009999685 -0.0000103731 0.0010000000**

***Задача 3***

#include <iostream>

#include <math.h>

using namespace std;

double func(double x1, double x2, double x3);

double pr(double x, double xi);

void MethodProGrad (double A, double B, double C, double a,double b,double y,double d,double e);

void print(double k,double x1,double x2,double x3,double dp, double d, double f)

{

cout<<k;

printf(" %.3lf",x1);

printf(" %.3lf",x2);

printf(" %.3lf",x3);

printf(" %.3lf",dp);

printf(" %.3lf",d);

printf(" %.3lf",f);

cout<<endl;

}

int main(int argc, const char \* argv[]) {

double A, B, C, a,b,y,d,e;

A=110;

B=180;

C=170;

a=3;

b=2;

y=2;

d=972;

e=0.001;

MethodProGrad (A, B, C, a, b, y, d, e);

return 0;

}

double func(double x1, double x2, double x3)

{

return -x1\*x2\*x3;

}

double pr(double x, double xi)

{

return -x\*xi;

}

void MethodProGrad (double A, double B, double C, double a,double b,double y,double d,double e)

{

int k;

double Sqrt, x10, x11, xp1, x20, x21, xp2, x30, x31, xp3, ak, pr1, pr2, pr3, t, f, dp;

k=0;

ak=1;

x10=A/2;

x20=B/2;

x30=C/2;

cout<<"Таблица значений:"<<endl;

cout<<"k x1 x2 x3 αx1+βx2+γx3 δ V"<<endl;

do {

k++;

pr1=pr(x20,x30);

pr2=pr(x10,x30);

pr3=pr(x10,x20);

Sqrt=sqrt(pow(pr1,2)+pow(pr2,2)+pow(pr3,2));

x11=x10-ak\*pr1/Sqrt;

x21=x20-ak\*pr2/Sqrt;

x31=x30-ak\*pr3/Sqrt;

if ((a\*x11+b\*x21+y\*x31)>d)

{

t=(d-a\*x11-b\*x21-y\*x31)/(pow(a,2)+pow(b,2)+pow(y,2));

x11=x10+ak\*t;

x21=x20+ak\*t;

x31=x30+ak\*t;

}

xp1=x10;

xp2=x20;

xp3=x30;

x10=x11;

x20=x21;

x30=x31;

f=fabs(func(x10,x20,x30));

dp=a\*x11+b\*x21+y\*x31;

print(k,x10,x20,x30,dp,d,f);

}

while ((fabs(xp1-x11)>=e)||(fabs(xp2-x21)>=e)||(fabs(xp3-x31)>=e)||((a\*x11+b\*x21+y\*x31)>d));

print(k,x10,x20,x30,dp,d,f);

cout<<endl<<"Ответ:"<<endl<<"V = "<<f<<endl<<"x1\* = "<<x10<<endl<<"x2\* = "<<x20<<endl<<"x3\* = "<<x30<<endl;

}

Таблица значений:

k x1 x2 x3 αx1+βx2+γx3 δ V

1 55.747 90.456 85.483 519.121 972.000 431064.904

2 56.491 90.915 85.969 523.242 972.000 441528.279

3 57.233 91.376 86.456 527.363 972.000 452141.447

4 57.972 91.839 86.945 531.485 972.000 462905.721

5 58.709 92.304 87.437 535.607 972.000 473822.406

6 59.443 92.771 87.929 539.729 972.000 484892.802

7 60.175 93.240 88.424 543.852 972.000 496118.204

8 60.904 93.711 88.921 547.974 972.000 507499.897

9 61.631 94.183 89.419 552.096 972.000 519039.165

10 62.356 94.657 89.918 556.219 972.000 530737.284

11 63.079 95.133 90.419 560.341 972.000 542595.525

12 63.799 95.611 90.922 564.463 972.000 554615.155

13 64.517 96.091 91.426 568.586 972.000 566797.436

14 65.234 96.572 91.932 572.708 972.000 579143.626

15 65.948 97.054 92.438 576.829 972.000 591654.979

16 66.660 97.538 92.947 580.951 972.000 604332.744

17 67.371 98.024 93.456 585.072 972.000 617178.168

18 68.079 98.511 93.967 589.193 972.000 630192.492

19 68.786 98.999 94.479 593.314 972.000 643376.957

20 69.491 99.489 94.992 597.434 972.000 656732.797

21 70.194 99.980 95.506 601.554 972.000 670261.246

22 70.895 100.472 96.022 605.674 972.000 683963.533

23 71.595 100.966 96.538 609.793 972.000 697840.885

24 72.293 101.461 97.056 613.912 972.000 711894.527

25 72.989 101.957 97.575 618.031 972.000 726125.680

26 73.684 102.454 98.094 622.149 972.000 740535.564

27 74.377 102.953 98.615 626.266 972.000 755125.395

28 75.069 103.452 99.137 630.384 972.000 769896.389

29 75.759 103.953 99.659 634.500 972.000 784849.757

30 76.447 104.455 100.183 638.617 972.000 799986.711

31 77.134 104.958 100.707 642.733 972.000 815308.459

32 77.820 105.462 101.232 646.848 972.000 830816.208

33 78.504 105.967 101.758 650.963 972.000 846511.163

34 79.187 106.473 102.285 655.077 972.000 862394.527

35 79.869 106.980 102.813 659.191 972.000 878467.501

36 80.549 107.487 103.341 663.305 972.000 894731.286

37 81.228 107.996 103.870 667.418 972.000 911187.081

38 81.906 108.506 104.400 671.531 972.000 927836.081

39 82.582 109.017 104.931 675.643 972.000 944679.484

40 83.258 109.528 105.462 679.754 972.000 961718.483

41 83.932 110.040 105.995 683.865 972.000 978954.271

42 84.605 110.554 106.527 687.976 972.000 996388.040

43 85.276 111.068 107.061 692.086 972.000 1014020.982

44 85.947 111.583 107.595 696.196 972.000 1031854.284

45 86.616 112.098 108.130 700.305 972.000 1049889.137

46 87.285 112.615 108.665 704.413 972.000 1068126.727

47 87.952 113.132 109.201 708.522 972.000 1086568.241

48 88.618 113.650 109.738 712.629 972.000 1105214.864

49 89.283 114.168 110.275 716.736 972.000 1124067.780

50 89.947 114.688 110.813 720.843 972.000 1143128.174

51 90.611 115.208 111.351 724.949 972.000 1162397.228

52 91.273 115.729 111.890 729.055 972.000 1181876.123

53 91.934 116.250 112.429 733.160 972.000 1201566.042

54 92.594 116.772 112.969 737.265 972.000 1221468.164

55 93.253 117.295 113.509 741.369 972.000 1241583.668

56 93.912 117.818 114.050 745.473 972.000 1261913.735

57 94.569 118.343 114.592 749.576 972.000 1282459.541

58 95.226 118.867 115.133 753.679 972.000 1303222.266

59 95.882 119.393 115.676 757.781 972.000 1324203.084

60 96.536 119.918 116.219 761.883 972.000 1345403.173

61 97.190 120.445 116.762 765.985 972.000 1366823.709

62 97.844 120.972 117.305 770.086 972.000 1388465.865

63 98.496 121.500 117.850 774.186 972.000 1410330.818

64 99.147 122.028 118.394 778.286 972.000 1432419.741

65 99.798 122.556 118.939 782.386 972.000 1454733.807

66 100.448 123.086 119.484 786.485 972.000 1477274.189

67 101.097 123.616 120.030 790.584 972.000 1500042.060

68 101.746 124.146 120.576 794.682 972.000 1523038.592

69 102.394 124.677 121.123 798.780 972.000 1546264.956

70 103.040 125.208 121.670 802.877 972.000 1569722.324

71 103.687 125.740 122.217 806.974 972.000 1593411.866

72 104.332 126.272 122.765 811.071 972.000 1617334.752

73 104.977 126.805 123.313 815.167 972.000 1641492.152

74 105.621 127.338 123.861 819.262 972.000 1665885.237

75 106.265 127.872 124.410 823.358 972.000 1690515.174

76 106.907 128.406 124.959 827.453 972.000 1715383.133

77 107.550 128.941 125.508 831.547 972.000 1740490.282

78 108.191 129.476 126.058 835.641 972.000 1765837.789

79 108.832 130.011 126.608 839.735 972.000 1791426.822

80 109.472 130.547 127.158 843.828 972.000 1817258.548

81 110.112 131.084 127.709 847.921 972.000 1843334.135

82 110.751 131.620 128.260 852.013 972.000 1869654.749

83 111.389 132.158 128.811 856.105 972.000 1896221.556

84 112.027 132.695 129.363 860.197 972.000 1923035.724

85 112.664 133.233 129.915 864.288 972.000 1950098.417

86 113.301 133.772 130.467 868.379 972.000 1977410.802

87 113.937 134.310 131.019 872.470 972.000 2004974.044

88 114.573 134.849 131.572 876.560 972.000 2032789.308

89 115.208 135.389 132.125 880.650 972.000 2060857.760

90 115.842 135.929 132.678 884.739 972.000 2089180.563

91 116.476 136.469 133.231 888.828 972.000 2117758.883

92 117.109 137.010 133.785 892.917 972.000 2146593.883

93 117.742 137.551 134.339 897.005 972.000 2175686.729

94 118.374 138.092 134.893 901.094 972.000 2205038.582

95 119.006 138.633 135.448 905.181 972.000 2234650.608

96 119.638 139.175 136.002 909.269 972.000 2264523.970

97 120.268 139.718 136.557 913.356 972.000 2294659.830

98 120.899 140.260 137.113 917.442 972.000 2325059.352

99 121.529 140.803 137.668 921.528 972.000 2355723.699

100 122.158 141.346 138.224 925.614 972.000 2386654.033

101 122.787 141.890 138.779 929.700 972.000 2417851.518

102 123.416 142.434 139.336 933.785 972.000 2449317.315

103 124.044 142.978 139.892 937.870 972.000 2481052.586

104 124.671 143.523 140.448 941.955 972.000 2513058.494

105 125.298 144.067 141.005 946.040 972.000 2545336.200

106 125.925 144.612 141.562 950.124 972.000 2577886.867

107 126.551 145.158 142.119 954.207 972.000 2610711.655

108 127.177 145.703 142.676 958.291 972.000 2643811.727

109 127.803 146.249 143.234 962.374 972.000 2677188.243

110 128.428 146.795 143.791 966.457 972.000 2710842.364

111 129.052 147.342 144.349 970.539 972.000 2744775.253

112 128.898 147.188 144.195 969.460 972.000 2735700.381

113 128.807 147.097 144.104 968.825 972.000 2730371.428

114 128.754 147.044 144.051 968.451 972.000 2727239.900

115 128.723 147.012 144.020 968.232 972.000 2725398.896

116 128.704 146.994 144.001 968.102 972.000 2724316.312

117 128.693 146.983 143.990 968.026 972.000 2723679.616

118 128.687 146.977 143.984 967.982 972.000 2723305.125

119 128.683 146.973 143.980 967.955 972.000 2723084.847

120 128.681 146.971 143.978 967.940 972.000 2722955.274

121 128.680 146.969 143.977 967.931 972.000 2722879.055

122 128.679 146.968 143.976 967.925 972.000 2722834.219

122 128.679 146.968 143.976 967.925 972.000 2722834.219

Ответ:

V = 2.72283e+06

x1\* = 128.679

x2\* = 146.968

x3\* = 143.976

***Задача 1***

#include<string>

#include <fstream>

#include <iostream>

#include <cstdlib>

#include <cstddef>

#include <math.h>

using namespace std;

double a, b, y, d;

void Grad(double x10, double x20, double e);

void print(double k, double x1, double x2, double f, double pr1, double pr2, double ak)

{

cout << k;

printf(" %.7lf", x1);

printf(" %.7lf", x2);

printf(" %.7lf", f);

printf(" %.10lf", pr1);

printf(" %.10lf", pr2);

printf(" %.10lf", ak);

cout << endl;

}

double func(double x1, double x2)

{

double f;

f = a\*pow((x2 + b - pow(x1, 2)), 2) + pow((y - x1), 2) + d\*pow(x1, 2);

return f;

}

double dfdx1(double x1, double x2)

{

double pr1;

pr1 = -4 \* x1\*a\*(x2 + b - pow(x1, 2)) - 2 \* (y - x1) + 2 \* d\*x1;

return pr1;

}

double dfdx2(double x1, double x2)

{

double pr2;

pr2 = 2 \* a\*(x2 + b - pow(x1, 2));

return pr2;

}

int main(int argc, const char \* argv[]) {

double x10, x20, e;

a=29;

b=15;

y=12;

d=2;

x10=2;

x20=0;

e=0.01;

int k;

int i;

double x1[100000], x2[100000], pr1[100000], pr2[100000], ak[100000], f[100000];

k = 0;

x1[k] = x10;

x2[k] = x20;

ak[k] = 1;

pr1[k] = dfdx1(x1[k], x2[k]);

pr2[k] = dfdx2(x1[k], x2[k]);

f[k] = func(x1[k], x2[k]);

do

{

x1[k + 1] = x1[k] - ak[k] \* pr1[k];

x2[k + 1] = x2[k] - ak[k] \* pr2[k];

f[k + 1] = func(x1[k + 1], x2[k + 1]);

while (fabs(f[k + 1])>fabs(f[k]))

{

ak[k] = ak[k] / 2;

x1[k + 1] = x1[k] - ak[k] \* pr1[k];

x2[k + 1] = x2[k] - ak[k] \* pr2[k];

f[k + 1] = func(x1[k + 1], x2[k + 1]);

}

k++;

ak[k] = ak[k - 1];

pr1[k] = dfdx1(x1[k], x2[k]);

pr2[k] = dfdx2(x1[k], x2[k]);

}

while ((fabs(pr1[k]>e)) || (fabs(pr2[k])>e));

setlocale(LC\_ALL, "Russian");

cout << "Таблица значений:" << endl;

cout << "k x1(k) x2(k) f(k) df/dx1 df2/dx2 ak" << endl;

cout << "Первые 10 итераций" << endl;

for (i = 0; i<10; i++)

print(i, x1[i], x2[i], f[i], pr1[i], pr2[i], ak[i]);

cout << "Последние 10 итераций" << endl;

for (i = k - 9; i <= k; i++)

print(i, x1[i], x2[i], f[i], pr1[i], pr2[i], ak[i]);

return 0;

}

Успешно time: 0 memory: 22264 signal:0

Таблица значений:

k x1(k) x2(k) f(k) df/dx1 df2/dx2 ak

Первые 10 итераций:

0 2.0000000 0.0000000 3617.0000000 -2564.0000000000 638.0000000000 0.0009765625

1 4.5039062 -0.6230469 1109.0660542 3089.7905952930 -342.6766662598 0.0004882812

2 2.9952194 -0.4557243 999.6996859 -1942.3200509079 323.2303023145 0.0004882812

3 3.9436179 -0.6135516 135.4145974 532.9104241729 -67.6090750349 0.0004882812

4 3.6834077 -0.5805393 117.3503345 -365.9245005520 49.4141530883 0.0004882812

5 3.8620818 -0.6046673 103.9090184 232.2870156224 -30.1799051280 0.0004882812

6 3.7486604 -0.5899311 99.8982619 -157.0145932968 20.7416168452 0.0004882812

7 3.8253277 -0.6000588 97.6684910 102.4275613657 -13.5250628969 0.0004882812

8 3.7753142 -0.5934548 96.8351809 -68.5922398962 8.9057653415 0.0004882812

9 3.8088065 -0.5978033 96.4282369 45.1603878914 -6.0790103430 0.0004882812

Последние 10 итераций:

58332 3.9864977 0.8919911 96.0005478 -0.0012527134 -0.0100039339 0.0004882812

58333 3.9864983 0.8919960 96.0005478 -0.0012526563 -0.0100034792 0.0004882812

58334 3.9864989 0.8920009 96.0005477 -0.0012525992 -0.0100030246 0.0004882812

58335 3.9864995 0.8920058 96.0005477 -0.0012525420 -0.0100025699 0.0004882812

58336 3.9865001 0.8920107 96.0005476 -0.0012524849 -0.0100021153 0.0004882812

58337 3.9865007 0.8920155 96.0005476 -0.0012524278 -0.0100016607 0.0004882812

58338 3.9865013 0.8920204 96.0005475 -0.0012523707 -0.0100012061 0.0004882812

58339 3.9865019 0.8920253 96.0005475 -0.0012523136 -0.0100007516 0.0004882812

58340 3.9865026 0.8920302 96.0005474 -0.0012522565 -0.0100002970 0.0004882812

58341 3.9865032 0.8920351 96.0005474 -0.0012521994 -0.0099998425 0.0004882812