



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)» (МГТУ  
им. Н.Э. Баумана)**

---

**ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)**

**КАФЕДРА «Информационная безопасность» (ИУ8)**

**Отчёт**

**по лабораторной работе № 3  
по дисциплине «Теория систем и системный анализ»**

**Тема: «Исследование алгоритма имитации отжига»**

**Вариант 3**

**Выполнила: Бакаев Ф.Б.,  
студент группы ИУ8-31**

**Проверила: Коннова Н.С.,  
доцент каф. ИУ8**

**г. Москва, 2020 г.**

## Цель работы

Изучение метода имитации отжига для поиска экстремума на примере унимодальной и мультимодальной функций одного переменного.

## Условие задачи

1. На интервале  $[0,3]$  задана унимодальная функция одного переменного  $f(x) = -\sqrt{x} \cdot \sin(x)$ . Используя метод имитации отжига осуществить поиск минимума  $f(x)$ .
2. При аналогичных исходных условиях осуществить поиск минимума  $f(x)$ , модулированной сигналом  $\sin(5x)$ , т.е. мультимодальной функции  $f(x) \cdot \sin(5x)$ .

## Графики заданных функций

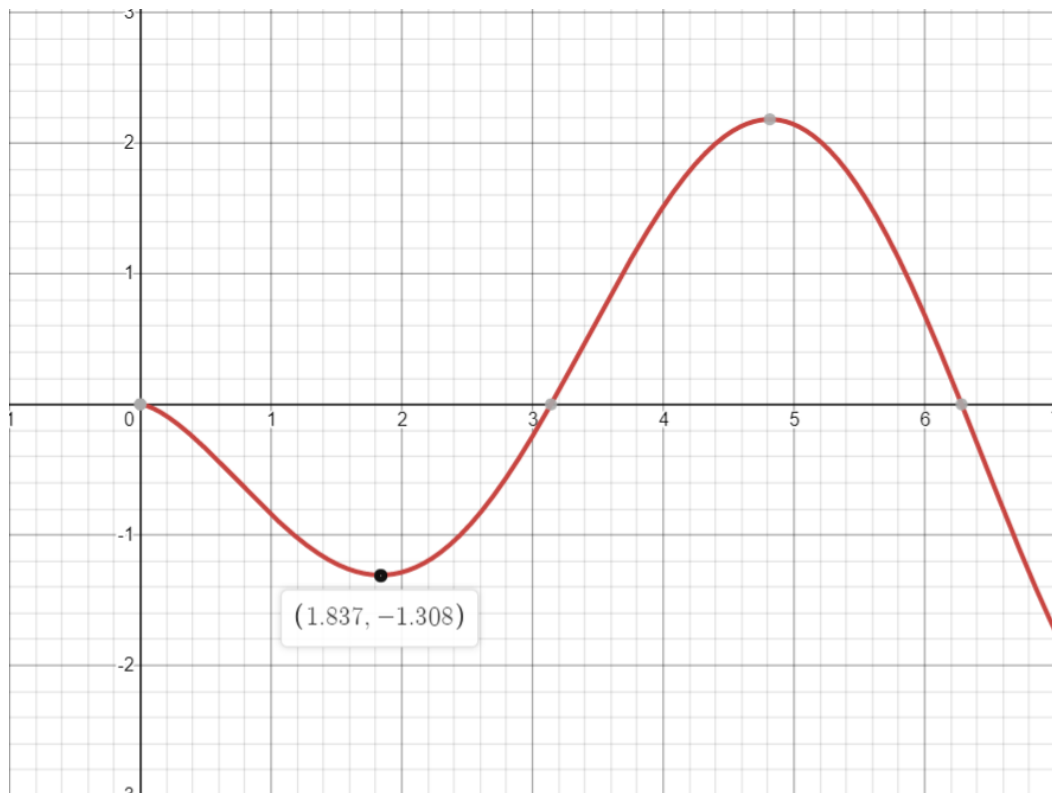


Рисунок 1 - График функции  $y = -\sqrt{x} \cdot \sin(x)$  на  $[0;3]$

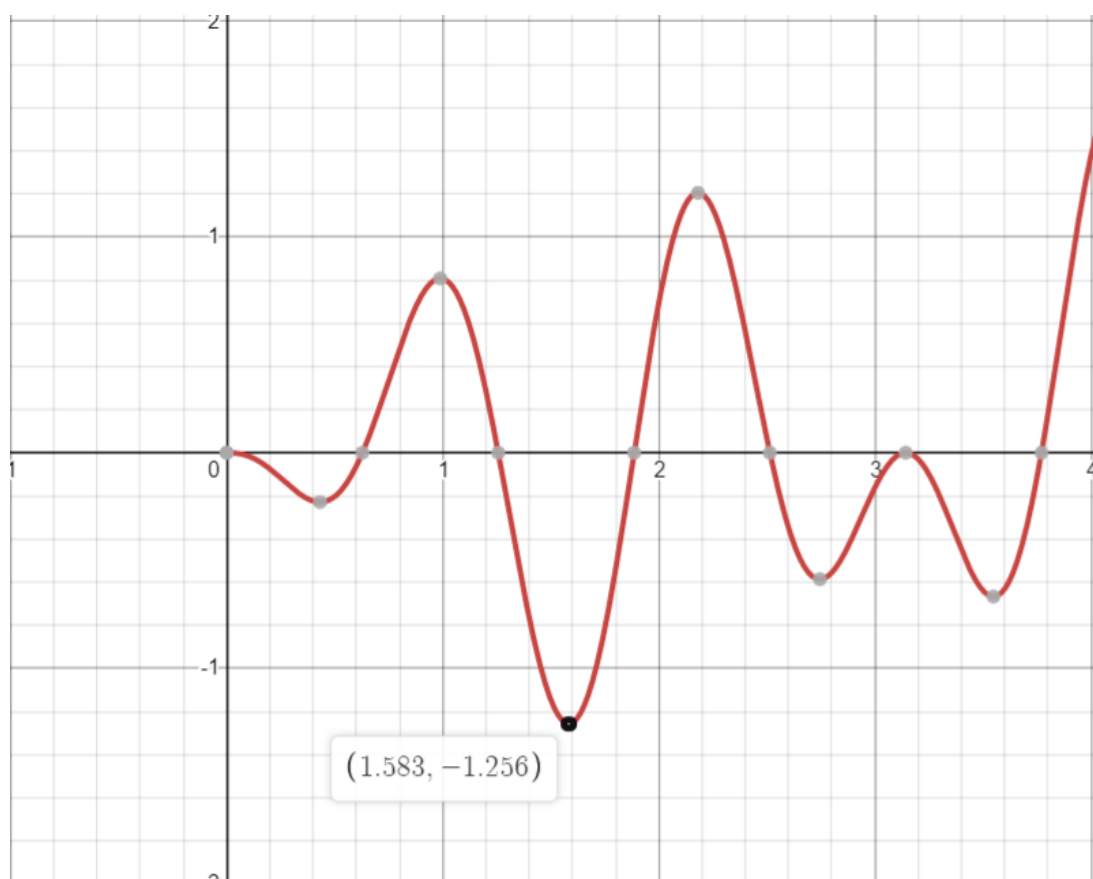


Рисунок 2 - График функции  $y = -\sqrt{x} \cdot \sin(x) \cdot \sin(5x)$  на  $[0; 3]$

### Имитация отжига для заданных функций

Имитация отжига для  $y = -\sqrt{x} \cdot \sin(x)$  на  $[0; 3]$

N	T	x	f(x)
1	10000	1.69076	-1.29095
2	9500	0.579913	-0.417276
3	9025	1.75503	-1.30236
4	8573.75	1.43962	-1.18953
5	8145.06	2.68789	-0.718582
6	7737.81	2.23981	-1.17398
7	7350.92	0.522324	-0.360562
8	6983.37	2.1315	-1.23642
9	6634.2	1.5406	-1.24065
10	6302.49	0.911985	-0.755121
11	5987.37	0.274209	-0.141797
12	5688	0.441939	-0.284323
13	5403.6	0.497696	-0.336796
14	5133.42	2.96558	-0.301554

15	4876.75	0.35725	−0.209016	
16	4632.91	0.0267342	−0.00437068	
17	4401.27	1.59499	−1.26256	
18	4181.2	1.71355	−1.29571	
19	3972.14	1.80529	−1.30684	
20	3773.54	1.8215	−1.30744	
21	3584.86	0.498703	−0.337761	
22	3405.62	1.35237	−1.13528	
23	3235.34	1.05637	−0.894774	
24	3073.57	1.82305	−1.30747	
25	2919.89	2.34996	−1.09071	
26	2773.9	1.55965	−1.24878	
27	2635.2	0.90585	−0.748989	
28	2503.44	2.18003	−1.21085	
29	2378.27	2.8677	−0.458037	
30	2259.36	1.61806	−1.27061	
31	2146.39	0.427015	−0.270636	
32	2039.07	0.705985	−0.545127	
33	1937.11	2.58672	−0.847326	
34	1840.26	0.628803	−0.466408	
35	1748.25	2.53096	−0.912198	
36	1660.83	2.53096	−0.912198	
37	1577.79	1.8345	−1.30762	
38	1498.9	1.17731	−1.00212	
39	1423.96	0.891842	−0.73494	
40	1352.76	0.0712302	−0.0189945	
41	1285.12	0.277871	−0.144598	
42	1220.87	2.03162	−1.27667	
43	1159.82	0.168645	−0.0689288	
44	1101.83	2.75637	−0.623857	
45	1046.74	0.827662	−0.669903	
46	994.403	0.81869	−0.660742	
47	944.682	2.07355	−1.2618	
48	897.448	2.51283	−0.932318	
49	852.576	1.45482	−1.19806	
50	809.947	0.616077	−0.453548	
51	769.45	1.40538	−1.1693	
52	730.977	1.37388	−1.14948	

53	694.428	2.23331	-1.17828	
54	659.707	0.324839	-0.181902	
55	626.722	1.15571	-0.983745	
56	595.386	2.20502	-1.19616	
57	565.616	1.8269	-1.30754	
58	537.335	1.71722	-1.2964	
59	510.469	0.454665	-0.296121	
60	484.945	1.27546	-1.08047	
61	460.698	2.40864	-1.03838	
62	437.663	2.96997	-0.294318	
63	415.78	1.03668	-0.876365	
64	394.991	0.506943	-0.345681	
65	375.241	1.47569	-1.20929	
66	356.479	0.190619	-0.0827208	
67	338.655	1.51442	-1.22866	
68	321.723	0.442488	-0.28483	
69	305.636	0.424726	-0.268551	
70	290.355	2.07868	-1.25978	
71	275.837	0.909146	-0.752285	
72	262.045	0.211127	-0.0962908	
73	248.943	2.04956	-1.27066	
74	236.496	0.4597	-0.30082	
75	224.671	2.46504	-0.983018	
76	213.437	1.74615	-1.30116	
77	202.765	0.574053	-0.411441	
78	192.627	2.45158	-0.996672	
79	182.996	1.42579	-1.18154	
80	173.846	0.466659	-0.307341	
81	165.154	2.19605	-1.20155	
82	156.896	1.21677	-1.03467	
83	149.051	1.70623	-1.29427	
84	141.599	2.04672	-1.27165	
85	134.519	2.16575	-1.21878	
86	127.793	0.36906	-0.21915	
87	121.403	2.50404	-0.941898	
88	115.333	0.105289	-0.0341013	
89	109.566	1.98895	-1.28879	
90	104.088	1.27866	-1.08287	

91	98.8836	2.84802	-0.488353	
92	93.9395	1.64864	-1.28011	
93	89.2425	1.03797	-0.877571	
94	84.7804	1.12494	-0.956949	
95	80.5413	2.54094	-0.900916	
96	76.5143	1.3683	-1.14584	
97	72.6886	0.815668	-0.657654	
98	69.0541	0.893399	-0.736504	
99	65.6014	2.21757	-1.18839	
100	62.3214	1.70183	-1.29336	
101	59.2053	0.58797	-0.425318	
102	56.245	2.51833	-0.926272	
103	53.4328	1.19297	-1.01519	
104	50.7611	1.5027	-1.22301	
105	48.2231	2.67049	-0.741694	
106	45.8119	2.98389	-0.271293	
107	43.5213	0.151524	-0.0587571	
108	41.3453	0.582202	-0.419558	
109	39.278	2.52913	-0.914252	
110	37.3141	1.88028	-1.30609	
111	35.4484	1.97284	-1.29258	
112	33.676	2.52647	-0.917222	
113	31.9922	0.329783	-0.185969	
114	30.3926	0.942198	-0.785128	
115	28.8729	2.82321	-0.525973	
116	27.4293	1.00894	-0.850044	
117	26.0578	0.420789	-0.264975	
118	24.7549	2.50386	-0.942097	
119	23.5172	2.124	-1.24002	
120	22.3413	1.80071	-1.30659	
121	21.2243	2.24165	-1.17276	
122	20.1631	0.433424	-0.276494	
123	19.1549	0.18302	-0.0778608	
124	18.1972	2.55788	-0.881438	
125	17.2873	0.631733	-0.469374	
126	16.4229	1.65963	-1.28319	
127	15.6018	0.0427564	-0.00883832	
128	14.8217	1.36354	-1.14271	

129	14.0806	2.25666	−1.16252	
130	13.3766	2.05844	−1.26749	
131	12.7078	1.63033	−1.27458	
132	12.0724	0.221656	−0.103504	
133	11.4687	0.605823	−0.443221	
134	10.8953	2.08866	−1.25572	
135	10.3505	0.871059	−0.71399	
136	9.83302	0.697287	−0.536211	
137	9.34136	1.59774	−1.26356	
138	8.8743	1.88604	−1.30565	
139	8.43058	0.480575	−0.320475	
140	8.00905	2.88913	−0.424584	
141	7.6086	2.08728	−1.25629	
142	7.22817	2.77441	−0.597953	
143	6.86676	1.00784	−0.848993	
144	6.52342	1.00784	−0.848993	
145	6.19725	1.37233	−1.14847	
146	5.88739	2.99405	−0.254374	
147	5.59302	1.87551	−1.3064	
148	5.31337	0.283181	−0.148688	
149	5.0477	2.79455	−0.568573	
150	4.79532	0.145299	−0.0551903	
151	4.55555	0.870052	−0.712971	
152	4.32777	0.681906	−0.520465	
153	4.11138	1.23215	−1.04698	
154	3.90581	0.605914	−0.443313	
155	3.71052	1.81243	−1.30715	
156	3.525	1.35484	−1.13694	
157	3.34875	1.79348	−1.30614	
158	3.18131	1.90417	−1.30394	
159	3.02224	2.48637	−0.960811	
160	2.87113	2.16272	−1.22042	
161	2.72758	1.69726	−1.29238	
162	2.5912	1.1254	−0.957353	
163	2.46164	2.21372	−1.1908	
164	2.33856	1.6654	−1.28473	
165	2.22163	2.71526	−0.681421	
166	2.11055	0.56682	−0.404258	

167	2.00502	2.09552	-1.25283	
168	1.90477	1.75384	-1.3022	
169	1.80953	1.0539	-0.892475	
170	1.71905	1.0539	-0.892475	
171	1.6331	1.83615	-1.30762	
172	1.55145	1.86114	-1.30714	
173	1.47387	2.41359	-1.03372	
174	1.40018	1.72811	-1.29834	
175	1.33017	1.72811	-1.29834	
176	1.26366	1.84411	-1.30757	
177	1.20048	2.18305	-1.20913	
178	1.14045	2.00333	-1.28504	
179	1.08343	2.92959	-0.360147	
180	1.02926	1.7076	-1.29454	
181	0.977798	0.917478	-0.760602	
182	0.928908	0.917478	-0.760602	
183	0.882462	2.55367	-0.88632	
184	0.838339	2.23295	-1.17852	
185	0.796422	0.464644	-0.30545	
186	0.756601	0.238044	-0.115048	
187	0.718771	1.92294	-1.30161	
188	0.682833	2.46001	-0.988159	
189	0.648691	1.34477	-1.13014	
190	0.616256	1.22694	-1.04283	
191	0.585444	1.39668	-1.16394	
192	0.556171	1.50362	-1.22345	
193	0.528363	1.50362	-1.22345	
194	0.501945	2.214	-1.19063	
195	0.476847	2.214	-1.19063	
196	0.453005	2.214	-1.19063	
197	0.430355	2.214	-1.19063	
198	0.408837	0.381695	-0.230132	
199	0.388395	2.3549	-1.08651	
200	0.368975	1.57375	-1.25449	
201	0.350527	1.82891	-1.30757	
202	0.333	2.86834	-0.457043	
203	0.31635	2.62691	-0.797839	
204	0.300533	1.96158	-1.29498	



205	0.285506	0.966369	-0.808873	
206	0.271231	1.51515	-1.22901	
207	0.257669	1.51515	-1.22901	
208	0.244786	1.51515	-1.22901	
209	0.232547	1.9884	-1.28893	
210	0.220919	1.9884	-1.28893	
211	0.209873	1.9884	-1.28893	
212	0.19938	1.9884	-1.28893	
213	0.189411	1.9884	-1.28893	
214	0.17994	1.59645	-1.26309	
215	0.170943	1.18555	-1.00903	
216	0.162396	2.12363	-1.24019	
217	0.154276	2.12363	-1.24019	
218	0.146562	2.12363	-1.24019	
219	0.139234	2.24201	-1.17251	
220	0.132272	2.24201	-1.17251	
221	0.125659	1.65844	-1.28286	
222	0.119376	1.65844	-1.28286	
223	0.113407	1.89254	-1.3051	
224	0.107737	1.89254	-1.3051	
225	0.10235	1.89254	-1.3051	
226	0.0972324	1.89254	-1.3051	
227	0.0923708	1.89254	-1.3051	
228	0.0877523	1.89254	-1.3051	
229	0.0833647	1.89254	-1.3051	
230	0.0791964	1.89254	-1.3051	
231	0.0752366	1.89254	-1.3051	
232	0.0714748	1.89254	-1.3051	
233	0.067901	1.89254	-1.3051	
234	0.064506	1.89254	-1.3051	
235	0.0612807	1.89254	-1.3051	
236	0.0582167	1.89254	-1.3051	
237	0.0553058	1.89254	-1.3051	
238	0.0525405	1.89254	-1.3051	
239	0.0499135	1.89254	-1.3051	
240	0.0474178	1.89254	-1.3051	
241	0.0450469	1.89254	-1.3051	
242	0.0427946	1.89254	-1.3051	

243	0.0406549	1.89254	-1.3051	
244	0.0386221	1.89254	-1.3051	
245	0.036691	1.72005	-1.29693	
246	0.0348565	1.64415	-1.2788	
247	0.0331136	1.64415	-1.2788	
248	0.031458	1.64415	-1.2788	
249	0.0298851	1.64415	-1.2788	
250	0.0283908	1.64415	-1.2788	
251	0.0269713	1.64415	-1.2788	
252	0.0256227	1.64415	-1.2788	
253	0.0243416	1.64415	-1.2788	
254	0.0231245	1.64415	-1.2788	
255	0.0219683	1.64415	-1.2788	
256	0.0208699	1.64415	-1.2788	
257	0.0198264	1.64415	-1.2788	
258	0.018835	1.64415	-1.2788	
259	0.0178933	1.64415	-1.2788	
260	0.0169986	1.64415	-1.2788	
261	0.0161487	1.64415	-1.2788	
262	0.0153413	1.64415	-1.2788	
263	0.0145742	1.64415	-1.2788	
264	0.0138455	1.64415	-1.2788	
265	0.0131532	1.64415	-1.2788	
266	0.0124956	1.64415	-1.2788	
267	0.0118708	1.64415	-1.2788	
268	0.0112772	1.64415	-1.2788	
269	0.0107134	1.64415	-1.2788	
270	0.0101777	1.64415	-1.2788	

-----  
Result: Xmin = 1.64415 Fmin = -1.2788

**Имитация отжига для  $y = -\sqrt{x} * \sin(x) * \sin(5x)$  на  $[0;3]$**

N	T	x	f(x)	
-----				
1	10000	0.765404	0.383688	
2	9500	2.83584	-0.506452	
3	9025	1.3065	-0.272222	

4	8573.75	0.0215155	-0.000338828	
5	8145.06	1.80465	-0.510727	
6	7737.81	2.35847	0.757349	
7	7350.92	0.42729	-0.228693	
8	6983.37	0.666982	0.0970604	
9	6634.2	0.0128178	-9.29378e-05	
10	6302.49	1.25376	0.0152967	
11	5987.37	1.9798	0.589511	
12	5688	0.194555	-0.0704758	
13	5403.6	2.43318	0.39566	
14	5133.42	2.07447	1.02427	
15	4876.75	1.59041	-1.25481	
16	4632.91	2.05689	0.960702	
17	4401.27	2.06851	1.00374	
18	4181.2	2.3332	0.865545	
19	3972.14	0.0932035	-0.0127669	
20	3773.54	2.60604	-0.368545	
21	3584.86	1.93356	0.312843	
22	3405.62	0.256355	-0.123055	
23	3235.34	1.65596	-1.16767	
24	3073.57	2.84371	-0.493286	
25	2919.89	0.824915	0.555131	
26	2773.9	2.94534	-0.278159	
27	2635.2	1.85995	-0.163005	
28	2503.44	2.76745	-0.580851	
29	2378.27	1.1026	0.652426	
30	2259.36	0.655904	0.0679096	
31	2146.39	0.467757	-0.221815	
32	2039.07	0.721641	0.252458	
33	1937.11	2.70684	-0.570757	
34	1840.26	0.319254	-0.177281	
35	1748.25	1.32517	-0.375204	
36	1660.83	0.240333	-0.10883	
37	1577.79	0.515366	-0.189367	
38	1498.9	2.92318	-0.328806	
39	1423.96	2.32762	0.888009	
40	1352.76	0.631916	0.00844628	
41	1285.12	1.36985	-0.615095	

42	1220.87	0.0112613	-6.72521e-05	
43	1159.82	0.342143	-0.19433	
44	1101.83	1.21412	0.21786	
45	1046.74	2.97784	-0.205439	
46	994.403	0.115726	-0.0214817	
47	944.682	0.567827	-0.120712	
48	897.448	0.743065	0.316539	
49	852.576	1.86096	-0.156468	
50	809.947	2.65612	-0.498142	
51	769.45	2.81726	-0.534203	
52	730.977	0.586962	-0.0871154	
53	694.428	1.93667	0.332316	
54	659.707	2.72744	-0.583239	
55	626.722	1.36637	-0.596978	
56	595.386	2.76388	-0.582424	
57	565.616	0.452284	-0.226557	
58	537.335	1.70861	-0.999337	
59	510.469	1.26612	-0.0508833	
60	484.945	1.62172	-1.23081	
61	460.698	1.73571	-0.882258	
62	437.663	0.765862	0.385059	
63	415.78	1.05078	0.762297	
64	394.991	2.38575	0.630647	
65	375.241	0.589526	-0.0822795	
66	356.479	0.210852	-0.0835666	
67	338.655	1.16825	0.425326	
68	321.723	0.212775	-0.0851592	
69	305.636	0.593005	-0.0755926	
70	290.355	1.93301	0.309396	
71	275.837	1.81289	-0.460878	
72	262.045	2.09204	1.07881	
73	248.943	2.05341	0.947092	
74	236.496	1.18958	0.333108	
75	224.671	2.50716	0.0287026	
76	213.437	1.77664	-0.672621	
77	202.765	0.598773	-0.0641954	
78	192.627	2.62801	-0.432256	
79	182.996	1.17512	0.396522	

80	173.846	0.555925	−0.139336	
81	165.154	2.68715	−0.549706	
82	156.896	1.72335	−0.938011	
83	149.051	1.32618	−0.380736	
84	141.599	2.12555	1.15639	
85	134.519	0.856685	0.636062	
86	127.793	0.780602	0.428968	
87	121.403	1.22291	0.174499	
88	115.333	2.68596	−0.548143	
89	109.566	2.13059	1.16497	
90	104.088	2.68816	−0.551002	
91	98.8836	1.1807	0.372469	
92	93.9395	2.7116	−0.574501	
93	89.2425	0.92581	0.766225	
94	84.7804	1.71172	−0.986871	
95	80.5413	1.06067	0.746339	
96	76.5143	2.23606	1.15645	
97	72.6886	2.2183	1.18247	
98	69.0541	0.600513	−0.0606828	
99	65.6014	0.817774	0.53565	
100	62.3214	2.73339	−0.585066	
101	59.2053	1.10352	0.64991	
102	56.245	0.328135	−0.18416	
103	53.4328	2.72231	−0.581032	
104	50.7611	0.60921	−0.0426061	
105	48.2231	1.96451	0.501378	
106	45.8119	2.06293	0.983544	
107	43.5213	0.238685	−0.107379	
108	41.3453	0.227332	−0.0974909	
109	39.278	1.06525	0.738144	
110	37.3141	2.09186	1.07829	
111	35.4484	1.18455	0.35559	
112	33.676	0.203528	−0.0775884	
113	31.9922	2.02722	0.834372	
114	30.3926	0.596118	−0.0694896	
115	28.8729	2.08206	1.04898	
116	27.4293	1.96469	0.502455	
117	26.0578	1.63198	−1.2159	

118	24.7549	0.2949	−0.157103	
119	23.5172	0.148869	−0.0387709	
120	22.3413	1.3921	−0.727664	
121	21.2243	2.90991	−0.358854	
122	20.1631	1.59288	−1.2541	
123	19.1549	2.03977	0.890578	
124	18.1972	1.11634	0.612662	
125	17.2873	1.16587	0.435108	
126	16.4229	0.149876	−0.0393766	
127	15.6018	2.42933	0.415079	
128	14.8217	0.18247	−0.0613086	
129	14.0806	0.931669	0.773579	
130	13.3766	0.63741	0.0215896	
131	12.7078	0.965728	0.802791	
132	12.0724	0.0274667	−0.000623109	
133	11.4687	1.31959	−0.344476	
134	10.8953	1.52358	−1.19875	
135	10.3505	1.81784	−0.430557	
136	9.83302	1.20322	0.270148	
137	9.34136	2.52794	−0.0670777	
138	8.8743	2.82696	−0.520323	
139	8.43058	2.82696	−0.520323	
140	8.00905	0.160405	−0.0459782	
141	7.6086	2.59066	−0.317925	
142	7.22817	1.16028	0.457634	
143	6.86676	0.852657	0.626391	
144	6.52342	1.8377	−0.306078	
145	6.19725	1.17356	0.403127	
146	5.88739	0.093112	−0.0127366	
147	5.59302	1.36711	−0.600802	
148	5.31337	2.57381	−0.257161	
149	5.0477	1.20569	0.258433	
150	4.79532	1.83477	−0.324667	
151	4.55555	1.83477	−0.324667	
152	4.32777	1.7478	−0.824161	
153	4.11138	1.78506	−0.625307	
154	3.90581	2.37953	0.6603	
155	3.71052	1.77187	−0.69879	

156	3.525	1.7683	-0.718134	
157	3.34875	1.7076	-1.00331	
158	3.18131	1.7076	-1.00331	
159	3.02224	1.03073	0.787345	
160	2.87113	0.801935	0.491128	
161	2.72758	1.02469	0.792979	
162	2.5912	2.31306	0.943767	
163	2.46164	2.60961	-0.3796	
164	2.33856	0.350566	-0.19998	
165	2.22163	1.88678	0.0118856	
166	2.11055	0.549059	-0.149272	
167	2.00502	1.60671	-1.24638	
168	1.90477	1.60671	-1.24638	
169	1.80953	2.89563	-0.390489	
170	1.71905	2.70464	-0.568853	
171	1.6331	2.70464	-0.568853	
172	1.55145	2.70464	-0.568853	
173	1.47387	1.88614	0.00770229	
174	1.40018	1.8508	-0.222233	
175	1.33017	0.794977	0.471087	
176	1.26366	0.794977	0.471087	
177	1.20048	0.512803	-0.191828	
178	1.14045	0.512803	-0.191828	
179	1.08343	0.244453	-0.112468	
180	1.02926	2.56603	-0.227323	
181	0.977798	0.373913	-0.213457	
182	0.928908	1.71996	-0.952613	
183	0.882462	1.41893	-0.85401	
184	0.838339	1.41893	-0.85401	
185	0.796422	2.98508	-0.189898	
186	0.756601	2.98508	-0.189898	
187	0.718771	0.522416	-0.18217	
188	0.682833	0.522416	-0.18217	
189	0.648691	2.51677	-0.0162195	
190	0.616256	2.51677	-0.0162195	
191	0.585444	0.697562	0.182053	
192	0.556171	0.694998	0.17471	
193	0.528363	0.613788	-0.0327544	

194	0.501945	2.91311	-0.351645	
195	0.476847	0.845241	0.608107	
196	0.453005	2.30766	0.963364	
197	0.430355	0.140904	-0.0341428	
198	0.408837	0.0277413	-0.000638762	
199	0.388395	1.93164	0.300767	
200	0.368975	1.24369	0.0683161	
201	0.350527	1.24955	0.0375773	
202	0.333	0.272011	-0.137024	
203	0.31635	1.63555	-1.2099	
204	0.300533	1.63555	-1.2099	
205	0.285506	1.63555	-1.2099	
206	0.271231	1.60643	-1.24659	
207	0.257669	1.71007	-0.993503	
208	0.244786	1.39732	-0.753128	
209	0.232547	1.39732	-0.753128	
210	0.220919	1.45976	-1.02042	
211	0.209873	1.45976	-1.02042	
212	0.19938	1.45976	-1.02042	
213	0.189411	1.45976	-1.02042	
214	0.17994	1.45976	-1.02042	
215	0.170943	1.45976	-1.02042	
216	0.162396	1.45976	-1.02042	
217	0.154276	1.45976	-1.02042	
218	0.146562	1.45976	-1.02042	
219	0.139234	1.45976	-1.02042	
220	0.132272	1.45976	-1.02042	
221	0.125659	1.45976	-1.02042	
222	0.119376	1.45976	-1.02042	
223	0.113407	1.45976	-1.02042	
224	0.107737	1.45976	-1.02042	
225	0.10235	1.45976	-1.02042	
226	0.0972324	1.45976	-1.02042	
227	0.0923708	1.45976	-1.02042	
228	0.0877523	1.45976	-1.02042	
229	0.0833647	1.45976	-1.02042	
230	0.0791964	1.45976	-1.02042	
231	0.0752366	1.45976	-1.02042	



232	0.0714748	1.45976	-1.02042	
233	0.067901	1.45976	-1.02042	
234	0.064506	1.45976	-1.02042	
235	0.0612807	1.45976	-1.02042	
236	0.0582167	1.6512	-1.17871	
237	0.0553058	1.6512	-1.17871	
238	0.0525405	1.6512	-1.17871	
239	0.0499135	1.6512	-1.17871	
240	0.0474178	1.6512	-1.17871	
241	0.0450469	1.6512	-1.17871	
242	0.0427946	1.6512	-1.17871	
243	0.0406549	1.6512	-1.17871	
244	0.0386221	1.6512	-1.17871	
245	0.036691	1.6512	-1.17871	
246	0.0348565	1.6512	-1.17871	
247	0.0331136	1.6512	-1.17871	
248	0.031458	1.6512	-1.17871	
249	0.0298851	1.6512	-1.17871	
250	0.0283908	1.64003	-1.20177	
251	0.0269713	1.64003	-1.20177	
252	0.0256227	1.64003	-1.20177	
253	0.0243416	1.64003	-1.20177	
254	0.0231245	1.64003	-1.20177	
255	0.0219683	1.64003	-1.20177	
256	0.0208699	1.64003	-1.20177	
257	0.0198264	1.64003	-1.20177	
258	0.018835	1.64003	-1.20177	
259	0.0178933	1.64003	-1.20177	
260	0.0169986	1.64003	-1.20177	
261	0.0161487	1.64003	-1.20177	
262	0.0153413	1.64003	-1.20177	
263	0.0145742	1.64003	-1.20177	
264	0.0138455	1.64003	-1.20177	
265	0.0131532	1.64003	-1.20177	
266	0.0124956	1.64003	-1.20177	
267	0.0118708	1.64003	-1.20177	
268	0.0112772	1.64003	-1.20177	
269	0.0107134	1.64003	-1.20177	

|270 |0.0101777 |1.64003 |-1.20177 |

-----

Result: Xmin = 1.64003 Fmin = -1.20177

## Выводы

Из полученных таблиц и графиков видно, что алгоритм имитации отжига является весьма эффективным алгоритмом случайного поиска глобального минимума. Применимость метода не зависит от того, является ли функция унимодальной или мультимодальной.

### Приложение. Исходный код программы

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <vector>
#include <string>

using std::cout;
using std::endl;

const auto a = 0.0;
const auto b = 3.0;

auto random(const double min, const double max) -> double {
    return (double)(rand()) / RAND_MAX * (max - min) + min;
}

auto f(const double x, const bool choice) -> double {
    if (choice == 0) {
        return -sqrt(x) * sin(x);
    }
    else if (choice == 1) {
        return f(x, 0) * sin(5 * x);
    }
}

auto Annealing(const bool choice) -> std::vector<std::pair<double, double>> {
    std::vector<std::pair<double, double>> result;
    auto x_min = random(a, b);
    auto t_max = 10000.0, t_min = 0.01;
    while (t_max > t_min) {
        auto x_i = random(a, b);
        auto difference = f(x_i, choice) - f(x_min, choice);
        auto probability = 0.0;
        if (difference <= 0) {
            x_min = x_i;
            probability = 1;
        }
        else {
            auto temp = random(0, 1);
            probability = exp(-difference / t_max);
            if (temp < probability) {
                x_min = x_i;
            }
        }
        result.emplace_back(t_max, x_min);
        t_max *= 0.95;
    }
    return result;
}

void Print(const std::vector<std::pair<double, double>>& table, const bool choice) {
    cout << std::string(44, '-') << endl;
    cout << " | " << std::setw(4) << std::left << "N";
    cout << " | " << std::setw(11) << std::left << "T";
    cout << " | " << std::setw(11) << std::left << "x";
```

```

cout << "|" << std::setw(13) << std::left << "f(x)";
cout << '|' << endl;
cout << std::string(44, '-') << endl;
for (size_t i = 0; i < table.size(); i++) {
    cout << "|";
    cout << std::setw(4) << std::left << i + 1;
    cout << '|' << std::setw(11) << std::left << table[i].first;
    cout << '|' << std::setw(11) << std::left << table[i].second;
    cout << '|' << std::setw(13) << std::left << f(table[i].second, choice);
    cout << '|' << endl;
}
cout << std::string(44, '-') << endl;
cout << "Result: " << "Xmin = " << table[table.size() - 1].second << " Fmin = " <<
f(table[table.size() - 1].second, choice) << endl;
}

int main() {
    setlocale(LC_ALL, "Russian");

    cout << "Алгоритм имитации отжига" << endl;
    cout << "Вариант №3:" << endl << endl;
    cout << "Функция  $-\sqrt{x} * \sin(x)$  для интервала  $[0, 3]$ " << endl;
    Print(Annealing(0), 0);
    cout << endl << endl << "Функция  $-\sqrt{x} * \sin(x) * \sin(5 * x)$  для интервала  $[0, 3]$ "
<< endl;
    Print(Annealing(1), 1);

    return 0;
}

```

## **Ответ на контрольный вопрос**

1. В чем состоит сущность метода имитации отжига? Какова область применимости данного метода?

Основная идея алгоритма случайного заключается в том, что итеративно выбирается точка, в которой может быть минимум и сравнивается значение функции в нём со значением в предыдущей точке. Если значение меньше, то выполняется переход в новую точку. Если же оно больше, то осуществляется переход в новую точку с некоторой вероятностью, которая зависит от значения параметра-температуры (чем меньше температура, тем меньше вероятность перехода). За счёт этого можно выходить из локальных минимумов.

В настоящее время метод имитации отжига применяется для решения многих оптимизационных задач – финансовых, компьютерной графики, комбинаторных, в телекоммуникационных сетях и многих других. Зачастую метод отжига используют для обучения нейронных сетей.