Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Новосибирский государственный технический университет»

Кафедра теоретической и прикладной информатики

Лабораторная работа № 2
по дисциплине «Методы оптимизации»

Методы спуска

Факультет:	ПМИ
Группа:	ПМИ-72
Вариант:	1
Студент:	Сычев Егор

Преподаватель: Постовалов Сергей Николаевич

Новосибирск
2020

1. Цель работы

Ознакомиться с методами поиска минимума функции n переменных в оптимизационных задачах без ограничений.

2. Задания

№	Вид работы
1.	Реализовать **два метода** поиска экстремума функции (разного порядка) Включить в реализуемый алгоритм собственную процедуру, реализующую одномерный поиск по направлению. Методы поиска для самостоятельной реализации выбираются студентом в зависимости от уровня сложности. Выбранные методы должны иметь разный порядок (например, метод Гаусса (нулевого порядка) - 1 балл и метод Ньютона (второго порядка) - 3 балла, итого 9 баллов).
2.	С использованием разработанного программного обеспечения исследовать алгоритмы на квадратичной функции $f(\overline{x}) = 100(x_2 - x_1)^2 + (1 - x_1)^2$, функции Розенброка $f(\overline{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ и на заданной в соответствии с вариантом тестовой функции, осуществляя спуск из различных исходных точек (не менее двух). Исследовать сходимость алгоритма, фиксируя точность определения минимума/максимума, количество итераций метода и количество вычислений функции в зависимости от задаваемой точности поиска. Результатом выполнения данного пункта должны быть выводы об объёме вычислений в зависимости от задаваемой точности и начального приближения.

Найти максимум заданной функции:

$$f(x, y) = 2exp\left\{-\left(\frac{x-1}{2}\right)^2 - \left(\frac{y-1}{1}\right)^2\right\} + 3exp\left\{-\left(\frac{x-2}{3}\right)^2 - \left(\frac{y-3}{2}\right)^2\right\}$$

3. Ход работы

a. Метод сопряженных градиентов в модификации Флетчера-Ривса

i. Квадратичная функция $f(\bar{x}) = 100(x_2 - x_1)^2 + (1 - x_1)^2$

$x_0 = (-0.1; 0.1)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	93	6	1,001	1,001	0
0,0001	139	5	1	1	0
0,00001	387	11	1	1	0
0,000001	296	7	1	1	0
0,0000001	369	7	1	1	0

$x_0 = (7; 7)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	108	6	1	1	0
0,0001	139	5	1,0001	1,0001	0
0,00001	249	7	1	1	0
0,000001	216	5	1	1	0
0,0000001	153	3	1	1	0

ii. Функция Розенброка $f(\bar{x}) = 100(x_2 - {x_1}^2)^2 + (1 - x_1)^2$

$x_0 = (-0.1; 0.1)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	649	40	1	1	0
0,0001	1433	59	1	1	0
0,00001	1147	33	1	1	0
0,000001	1436	33	0,999999	0,999998	0
0,0000001	1382	27	0,9999999	0,9999998	0

$x_0 = (7; 7)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	1832	133	1	1,001	0
0,0001	1441	68	1	1	0
0,00001	2377	79	1,00001	1,00001	0
0,000001	1439	35	1	0,999999	0
0,0000001	2527	51	0,9999999	0,9999998	0

iii. Функция из варианта (поиск максимума)

$x_0 = (-0.1; 0.1)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	145	4	1,263	1,333	3,169
0,0001	222	5	1,2625	1,3343	3,1693
0,00001	319	6	1,26303	1,3344	3,16932
0,000001	366	6	1,263035	1,334396	3,169317
0,0000001	414	6	1,263035	1,3343957	3,1693172

$x_0 = (7; 7)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	131	4	1,965	2,882	3,035
0,0001	264	6	1,9672	2,8861	3,0351
0,00001	311	6	1,96715	2,88611	3,03506
0,000001	359	6	1,967151	2,886115	3,035064
0,0000001	406	6	1,9671515	2,8861149	3,0350635

b. Метод Бройдена (переменной метрики)

i. Квадратичная функция $f(\bar{x}) = 100(x_2 - x_1)^2 + (1 - x_1)^2$

$x_0 = (-0.1; 0.1)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	57	3	1	1	0
0,0001	80	3	1	1	0
0,00001	104	3	1	1	0
0,000001	91	2	1	1	0
0,0000001	153	3	1	1	0

$x_0 = (7; 7)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	106	4	1	1	0
0,0001	121	4	1	1	0
0,00001	155	4	1	1	0
0,000001	184	4	1	1	0
0,0000001	220	4	1	1	0

ii. Функция Розенброка $f(\bar{x}) = 100(x_2 - {x_1}^2)^2 + (1 - x_1)^2$

$x_0 = (-0.1; 0.1)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	487	25	1	1	0
0,0001	680	25	1	1	0
0,00001	887	25	1	1	0
0,000001	1080	25	1	1	0
0,0000001	1281	25	1	1	0

$x_0 = (7; 7)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	1329	69	1	1	0
0,0001	873	33	1	1	0
0,00001	1058	31	1	1	0
0,000001	1381	33	1	1	0
0,0000001	2436	49	1	1	0

iii. Функция из варианта (поиск максимума)

$x_0 = (-0.1; 0.1)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	145	4	1,263	1,333	3,169
0,0001	222	5	1,2625	1,3343	3,1693
0,00001	319	6	1,26304	1,3344	3,16932
0,000001	366	6	1,263035	1,334396	3,169317
0,0000001	414	6	1,263035	1,3343959	3,1693172

$x_0 = (7; 7)$

epsilon	function calls	iterations	x	y	f(x, y)
0,001	131	4	1,963	2,878	3,035
0,0001	264	6	1,9672	2,8861	3,0351
0,00001	311	6	1,96715	2,88612	3,03506
0,000001	359	6	1,967151	2,886115	3,035064
0,0000001	406	6	1,967152	2,886115	3,035064

4. Анализ сходимости

$\varepsilon = 10^{-3}$

$x_0 = (3; -3)$

a. Метод сопряженных градиентов в модификации Флетчера-Ривса

i. Квадратичная функция $f(\bar{x}) = 100(x_2 - x_1)^2 + (1 - x_1)^2$

i	xi	yi	f(x, y)	si1	si2	lambdai	|yi-y(i-1)|	|yi-y(i-1)|	|fi-f(i-1)|	angle((xi, yi), si)"	gi1	gi2
0	3	-3	3604	1204	-1200	0	0	0	0	0,0016638919752473356"	1204	-1200
1	0,1706	-0,18	12,97994	72,46274	-74,1082	-0,00235	2,8294	2,82	3591,02	0,015578549011534113"	68,4612	-70,12
2	0,00031256	-0,005845621	1,003167	-0,76774	-1,23164	-0,00235	0,170287	0,174154	11,97677	0,6108290849775809"	-0,76774	-1,23164
3	0,03676524	0,052633184	0,953	-18,2516	-17,9246	-0,04748	0,036453	0,058479	0,050167	2,956887098313727"	-5,10006	3,173589
4	1,00110953	0,99970128	0,0002	0,283869	-0,28165	-0,05284	0,964344	0,947068	0,952801	1,5661685721268461"	0,283869	-0,28165
5	1,000442438	1,000363158	8,24E-07	0,017685	-0,01679	-0,00235	0,000667	0,000662	0,000199	1,5448829744144432"	0,016741	-0,01586
6	1,000400879	1,00040262	1,61E-07	0,000454	0,000348	-0,00235	4,16E-05	3,95E-05	6,63E-07	0,13052339023200682"	0,000454	0,000348

ii. Функция Розенброка $f(\bar{x}) = 100(x_2 - {x_1}^2)^2 + (1 - x_1)^2$

i	xi	yi	f(x, y)	si1	si2	lambdai	|yi-y(i-1)|	|yi-y(i-1)|	|fi-f(i-1)|	angle((xi, yi), si)"	gi1	gi2
0	3	-3	14404	14404	-2400	0	0	0	0	0,6202945188567891"	14404	-2400
1	1,399556	-2,73333	2201,73	3153,381	-1026,03	-0,00011	1,600444	0,266667	12202,27	0,7829971579917454"	2627,535	-938,418

2	-0,7027	-2,04931	649,6326	-718,217	-508,619	-0,00067	2,102254	0,684023	1552,097	0,6242774987335835"	-718,217	-508,619
3	0,374627	-1,28638	203,9461	95,15218	-368,48	-0,0015	1,077326	0,762929	445,6865	0,030678947968208633"	212,5456	-285,345
4	0,056462	-0,05428	1,220491	-0,58923	-11,4931	-0,00334	0,318165	1,232104	202,7257	0,8563466961705548"	-0,58923	-11,4931
5	0,059413	0,003281	0,88471	-1,89093	-0,35508	-0,00501	0,002951	0,057559	0,335781	3,01115008330035"	-1,87527	-0,04969
6	0,25125	0,039305	0,617375	0,896622	-4,76442	-0,10145	0,191837	0,036023	0,267335	1,5399589398663636"	0,896622	-4,76442
7	0,24744	0,059553	0,566627	-1,26681	-0,72103	-0,00425	0,003811	0,020249	0,050748	2,860341243262149"	-1,33954	-0,3346
8	0,426282	0,161344	0,370655	2,326278	-4,07443	-0,14117	0,178842	0,101791	0,195972	1,4138429129170191"	2,326278	-4,07443
9	0,419703	0,172867	0,337823	-0,52455	-0,80524	-0,00283	0,006579	0,011523	0,032832	2,5388812395676035"	-0,60937	-0,65668
10	0,437052	0,199499	0,32411	-2,60922	1,69696	-0,03307	0,017349	0,026632	0,013713	2,1367385521461024"	-2,60922	1,69696
11	0,444431	0,1947	0,309452	-0,79582	-0,44295	-0,00283	0,007379	0,004799	0,014658	3,046606764084174"	-0,60999	-0,56381
12	0,473752	0,21102	0,29495	1,490858	-2,68427	-0,03684	0,029321	0,01632	0,014502	1,482852564638558"	1,490858	-2,68427
13	0,469536	0,218611	0,281736	-0,61088	-0,55435	-0,00283	0,004216	0,007591	0,013215	2,840407227143663"	-0,71297	-0,37054

14	0,577864	0,316916	0,207136	3,087726	-3,40218	-0,17733	0,108329	0,098305	0,0746	1,3354476671842337"	3,087726	-3,40218
15	0,570608	0,324912	0,184424	-0,62803	-0,21912	-0,00235	0,007256	0,007995	0,022712	2,959666292376932"	-0,70306	-0,13646
16	0,576274	0,326888	0,182251	0,351969	-1,04067	-0,00902	0,005666	0,001977	0,002173	1,7606456832732593"	0,351969	-1,04067
17	0,575279	0,329831	0,180512	-0,47605	-0,56892	-0,00283	0,000995	0,002943	0,001739	2,788134301268306"	-0,59311	-0,22279
18	0,709079	0,489737	0,101682	3,121376	-2,61129	-0,28107	0,133801	0,159906	0,078831	1,3010863368334984"	3,121376	-2,61129
19	0,704397	0,493654	0,088017	0,169865	-0,54664	-0,0015	0,004682	0,003917	0,013665	1,8807887267256485"	0,119253	-0,5043
20	0,703998	0,494939	0,087663	-0,40201	-0,13494	-0,00235	0,000399	0,001285	0,000354	2,8526726066556436"	-0,40201	-0,13494
21	0,70587	0,495567	0,087234	-0,53952	-0,77524	-0,00466	0,001872	0,000628	0,000429	2,790901569527316"	0,169975	-0,53709
22	0,808415	0,642914	0,047983	3,050911	-2,12396	-0,19007	0,102545	0,147348	0,039251	1,280007694830577"	3,050911	-2,12396
23	0,803838	0,6461	0,03848	-0,34346	-0,03269	-0,0015	0,004576	0,003186	0,009503	2,5594465036441436"	-0,37445	-0,01112
24	0,804645	0,646177	0,038327	0,020321	-0,25541	-0,00235	0,000807	7,68E-05	0,000153	2,1680040447730304"	0,020321	-0,25541
25	0,804569	0,647135	0,038197	-0,29388	-0,46284	-0,00375	7,62E-05	0,000958	0,000129	2,8138954167225507"	-0,32758	-0,03933

26	0,873678	0,755976	0,021342	2,311785	-1,4676	-0,23516	0,069109	0,108841	0,016855	1,2789390897879747"	2,311785	-1,4676
27	0,871182	0,757561	0,016789	0,269255	-0,30485	-0,00108	0,002497	0,001585	0,004553	1,5630718102313572"	0,229029	-0,27931
28	0,870891	0,75789	0,016701	-0,06288	-0,11215	-0,00108	0,000291	0,000329	8,86E-05	2,797933989767279"	-0,06288	-0,11215
29	0,952865	0,904095	0,003709	-8,08637	-17,6474	-1,30367	0,081974	0,146205	0,012991	2,7596035975533137"	1,375725	-0,77136
30	0,971868	0,945566	0,000899	-0,46021	0,207819	-0,00235	0,019003	0,041472	0,00281	1,9457477781365775"	-0,46021	0,207819
31	0,972365	0,945342	0,000766	0,002041	-0,02958	-0,00108	0,000497	0,000224	0,000133	2,2732119668595248"	0,003727	-0,03034
32	0,972356	0,945468	0,000764	-0,05174	-0,00182	-0,00425	8,67E-06	0,000126	1,81E-06	2,405459646812532"	-0,05174	-0,00182
33	0,972434	0,94547	0,000762	-0,01376	-0,03216	-0,0015	7,76E-05	2,74E-06	1,82E-06	2,7463528559370283"	0,006061	-0,03146
34	0,974516	0,950339	0,000692	-0,30683	0,131277	-0,15136	0,002082	0,004868	6,99E-05	1,9644735811167877"	-0,30683	0,131277
35	0,974848	0,950197	0,000634	-0,00098	-0,02546	-0,00108	0,000331	0,000142	5,81E-05	2,381976107395055"	0,000921	-0,02627
36	0,974854	0,950353	0,000632	-0,05553	0,002688	-0,00615	6,05E-06	0,000157	2,01E-06	2,32055258094778"	-0,05553	0,002688
37	0,974914	0,95035	0,00063	-0,01817	-0,02082	-0,00108	6E-05	2,9E-06	1,9E-06	3,0608985366660644"	-0,00868	-0,02128

38	0,975168	0,950642	0,000626	0,071522	-0,06214	-0,01398	0,000254	0,000291	4,16E-06	1,48795467546479"	0,071522	-0,06214
39	0,975091	0,950709	0,000621	-0,00935	-0,02225	-0,00108	7,72E-05	6,71E-05	4,95E-06	2,7414357441665578"	-0,01357	-0,01859
40	0,976592	0,954281	0,000578	-0,26122	0,109771	-0,16053	0,001501	0,003572	4,33E-05	1,969928554139008"	-0,26122	0,109771
41	0,976874	0,954162	0,000536	-0,00095	-0,02336	-0,00108	0,000282	0,000119	4,18E-05	2,3850901472392176"	0,000952	-0,02416
42	0,97688	0,954306	0,000535	-0,05072	0,002294	-0,00615	5,85E-06	0,000144	1,72E-06	2,3226900368630004"	-0,05072	0,002294
43	0,976935	0,954304	0,000533	-0,01659	-0,01921	-0,00108	5,48E-05	2,48E-06	1,58E-06	3,056725794543317"	-0,00782	-0,01961
44	0,977174	0,954581	0,000529	0,067116	-0,0577	-0,01443	0,000239	0,000277	3,62E-06	1,4838118239802047"	0,067116	-0,0577
45	0,977102	0,954643	0,000525	-0,00892	-0,02021	-0,00108	7,25E-05	6,23E-05	4,29E-06	2,760008661989359"	-0,01276	-0,01691
46	0,979517	0,960119	0,000464	-0,30159	0,133037	-0,27095	0,002416	0,005476	6,13E-05	1,9507509468909126"	-0,30159	0,133037
47	0,979843	0,959975	0,000408	0,003828	-0,02264	-0,00108	0,000326	0,000144	5,61E-05	2,1784175396506935"	0,005422	-0,02334
48	0,979832	0,960039	0,000407	-0,02801	-0,00629	-0,00283	1,08E-05	6,4E-05	8,26E-07	2,587391976693573"	-0,02801	-0,00629
49	0,979886	0,960051	0,000406	-0,01505	-0,03028	-0,00192	5,38E-05	1,21E-05	7,09E-07	2,807152551183881"	0,00867	-0,02495

50	0,999278	0,999076	2,75E-05	-0,20912	0,103912	-1,28881	0,019392	0,039024	0,000379	1,8951266029437075"	-0,20912	0,103912
51	0,999504	0,998964	4,41E-07	0,015274	-0,00814	-0,00108	0,000226	0,000112	2,71E-05	1,2748687994650614"	0,016633	-0,00882
52	0,999487	0,998972	2,63E-07	-9,9E-05	-0,00046	-0,00108	1,65E-05	8,79E-06	1,77E-07	2,56626495869736"	-9,9E-05	-0,00046

iii. Функция из варианта (поиск максимума)

i	xi	yi	f(x, y)	si1	si2	lambdai	|yi-y(i-1)|	|yi-y(i-1)|	|fi-f(i-1)|	angle((xi, yi), si)"	gi1	gi2
0	3	-3	0,000331	671,1856	-9056,86	0	0	0	0	0,7114253932326264"	671,1856	-9056,86
1	2,463052	4,245487	1,987733	0,051773	0,313329	-0,0008	0,536948	7,245487	3017,193	0,36194438688258684"	0,051773	0,31334
2	2,237182	2,878545	3,01035	0,020008	-0,00331	-4,36265	0,225869	1,366942	0,170898	1,074263524349068"	0,020008	-0,00331
3	1,979319	2,921256	3,034459	0,00123	0,003466	-12,8883	0,257863	0,042711	0,002639	0,25453450527061455"	0,000592	0,003571
4	1,966868	2,886164	3,035063	-2,1E-05	7,61E-06	-10,1252	0,012452	0,035092	6,56E-05	1,8257789733903813"	-2,1E-05	7,61E-06

b. Метод Бройдена (переменной метрики)

i. Квадратичная функция $f(\bar{x}) = 100(x_2 - x_1)^2 + (1 - x_1)^2$

i	xi	yi	f(x, y)	si1	si2	lambdai	|yi-y(i-1)|	|yi-y(i-1)|	|fi-f(i-1)|	angle((xi, yi), si)	gi1	gi2	etai11	etai12	etai21	etai22
0	3,000	-3,000	3604,000	1204,000	-1200,000	0,000	0,000	0,000	0,000	0,002	1204,000	-1200,000	1,000	0,000	0,000	1,000
1	0,171	-0,180	12,980	1204,000	-1200,000	-0,002	2,829	2,820	3591,020	0,028	68,461	-70,120	0,499	0,499	0,499	0,504

2	0,997	0,997	0,000	-0,827	-1,177	-1,000	0,827	1,177	12,980	2,968	0,003	-0,008	1,000	0,000	0,000	1,000
3	0,997	0,997	0,000	0,003	-0,008	-0,003	0,000	0,000	0,000	1,989	-0,003	-0,002	0,499	0,499	0,499	0,504
4	1,000	1,000	0,000	-0,003	-0,003	-1,002	0,003	0,003	0,000	3,140	0,000	0,000	1,000	0,000	0,000	1,000

ii. **Функция Розенброка** $f(\bar{x}) = 100(x_2 - {x_1}^2)^2 + (1 - x_1)^2$

i	xi	yi	f(x, y)	si1	si2	lambdai	|yi-y(i-1)|	|yi-y(i-1)|	|fi-f(i-1)|	angle((xi, yi), si)	gi1	gi2	etai11	etai12	etai21	etai22
0	3,000	-3,000	14404,000	14404,000	-2400,000	0,000	0,000	0,000	0,000	0,620	14404,000	-2400,000	1,000	0,000	0,000	1,000
1	1,400	-2,733	2201,730	14404,000	-2400,000	0,000	1,600	0,267	12202,270	0,932	2627,535	-938,418	0,015	0,122	0,122	0,985
2	2,506	6,229	2,542	-74,471	-603,086	-0,015	1,107	8,962	2199,187	2,882	55,456	-10,463	1,000	0,000	0,000	1,000
3	2,500	6,230	2,291	55,456	-10,463	0,000	0,006	0,001	0,251	1,376	23,303	-4,060	0,038	0,191	0,191	0,962
4	2,249	5,019	1,708	0,115	0,557	-2,176	0,251	1,211	0,583	0,217	37,176	-7,710	1,000	0,000	0,000	1,000
5	2,245	5,020	1,586	37,176	-7,710	0,000	0,004	0,001	0,122	1,355	19,669	-3,827	0,047	0,211	0,211	0,953
6	2,014	4,021	1,144	0,118	0,510	-1,958	0,231	0,999	0,441	0,237	29,580	-6,842	1,000	0,000	0,000	1,000
7	2,010	4,021	1,062	29,580	-6,842	0,000	0,003	0,001	0,083	1,334	18,282	-4,044	0,058	0,233	0,233	0,942
8	1,792	3,181	0,717	0,118	0,453	-1,856	0,218	0,840	0,344	0,259	23,127	-6,011	1,000	0,000	0,000	1,000
9	1,789	3,181	0,664	23,127	-6,011	0,000	0,003	0,001	0,054	1,313	16,027	-4,037	0,072	0,258	0,258	0,928
10	1,586	2,490	0,408	0,114	0,387	-1,788	0,203	0,692	0,255	0,281	17,359	-5,104	1,000	0,000	0,000	1,000

11	1,574	2,493	0,351	17,359	-5,104	-0,001	0,012	0,003	0,058	1,293	-7,956	2,892	0,091	0,287	0,287	0,909
12	1,464	2,131	0,232	0,105	0,346	-1,047	0,110	0,362	0,118	0,307	8,510	-2,589	1,000	0,000	0,000	1,000
13	1,461	2,132	0,214	8,510	-2,589	0,000	0,003	0,001	0,019	1,265	2,705	-0,610	0,105	0,305	0,305	0,896
14	1,333	1,761	0,133	0,097	0,279	-1,327	0,128	0,371	0,080	0,314	8,671	-3,003	1,000	0,000	0,000	1,000
15	1,326	1,764	0,110	8,671	-3,003	-0,001	0,007	0,002	0,024	1,260	-2,441	1,166	0,124	0,329	0,329	0,877
16	1,234	1,514	0,063	0,081	0,220	-1,135	0,092	0,250	0,046	0,332	4,994	-1,833	1,000	0,000	0,000	1,000
17	1,230	1,516	0,053	4,994	-1,833	-0,001	0,004	0,001	0,010	1,241	-0,596	0,429	0,141	0,348	0,348	0,859
18	1,147	1,307	0,027	0,065	0,162	-1,288	0,084	0,208	0,027	0,338	3,690	-1,482	1,000	0,000	0,000	1,000
19	1,144	1,308	0,021	3,690	-1,482	-0,001	0,003	0,001	0,006	1,234	0,041	0,108	0,160	0,366	0,366	0,841
20	1,074	1,148	0,008	0,046	0,105	-1,517	0,070	0,160	0,013	0,341	2,282	-0,993	1,000	0,000	0,000	1,000
21	1,072	1,149	0,005	2,282	-0,993	-0,001	0,002	0,001	0,003	1,231	0,254	-0,051	0,178	0,382	0,382	0,823
22	1,020	1,039	0,001	0,026	0,055	-2,009	0,052	0,110	0,004	0,338	0,961	-0,451	1,000	0,000	0,000	1,000
23	1,019	1,039	0,000	0,961	-0,451	-0,001	0,001	0,000	0,001	1,234	-0,104	0,070	0,194	0,394	0,394	0,807
24	1,004	1,007	0,000	0,007	0,015	-2,070	0,015	0,032	0,000	0,336	0,225	-0,108	1,000	0,000	0,000	1,000
25	1,004	1,007	0,000	0,225	-0,108	-0,001	0,000	0,000	0,000	1,235	-0,018	0,013	0,199	0,398	0,398	0,802
26	1,000	1,001	0,000	0,001	0,003	-2,307	0,003	0,007	0,000	0,331	0,031	-0,015	1,000	0,000	0,000	1,000
27	1,000	1,001	0,000	0,031	-0,015	-0,001	0,000	0,000	0,000	1,240	-0,002	0,002	0,200	0,399	0,399	0,801

28	1,000	1,000	0,000	0,000	0,000	-2,184	0,000	0,001	0,000	0,333	0,003	-0,001	1,000	0,000	0,000	1,000
29	1,000	1,000	0,000	0,003	-0,001	-0,001	0,000	0,000	0,000	1,238	0,000	0,000	0,200	0,399	0,399	0,801

iii. Функция из варианта (поиск максимума)

i	xi	yi	f(x, y)	si1	si2	lambdai	|yi-y(i-1)|	|yi-y(i-1)|	|fi-f(i-1)|	angle((xi, yi), si)	gi1	gi2	etai11	etai12	etai21	etai22
0	3,000	-3,000	0,000	671,186	-9056,858	0,000	0,000	0,000	0,000	0,711	671,186	-9056,858	1,000	0,000	0,000	1,000
1	2,463	4,245	1,988	671,186	-9056,858	-0,001	0,537	7,245	3017,193	2,542	0,052	0,313	0,995	0,074	0,074	0,006
2	1,792	4,194	2,091	0,075	0,006	-9,000	0,671	0,052	0,025	1,090	-0,022	0,285	1,000	0,000	0,000	1,000
3	1,894	2,875	3,033	-0,022	0,285	-4,618	0,102	1,318	0,149	0,660	-0,005	0,000	1,024	-0,297	-0,297	4,594
4	1,967	2,879	3,035	-0,005	0,000	-13,775	0,073	0,004	0,000	2,226	0,000	-0,001	1,000	0,000	0,000	1,000

5. Текст программы

descent_methods.rs

```
use nalgebra::allocator::Allocator;
use nalgebra::DefaultAllocator;
use nalgebra::DimName;
use nalgebra::VectorN;

use super::one_dimension_searchers::minimize;

pub fn conjugate_gradients<D>(
    f: &dyn Fn(&VectorN<f64, D>) -> f64,
    df: &dyn Fn(&VectorN<f64, D>) -> VectorN<f64, D>,
    mut x: VectorN<f64, D>,
    eps: f64,
    rev: bool,
) -> (VectorN<f64, D>, i32, i32, String)
where
    D: DimName,
    DefaultAllocator: Allocator<f64, D>,
{
    let mut iter = 0;
    let mut func_calls = 1;
    let mut result = String::new();

    let precision = -eps.log10().round() as usize;
    result.push_str(&for-
mat!("\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"
{}\";\"{}\";\n",
"i", "xi", "yi", "f(x, y)", "si1", "si2", "lambdai", "|yi-y(i-1)|", "|yi-y(i-
1)|", "|fi-f(i-1)|",
"angle((xi, yi), si)", "gi1", "gi2"));

    let mut g = df(&x);
    #[allow(non_snake_case)]
    let mut S = g.clone();

    result.push_str(&for-
mat!("\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"
{}\";\"{}\";\n",
    iter,
    x[0],
    x[1],
    if rev { 1./ f(&x) } else {f(&x)},
    S[0],
    S[1],
    0,
    0,
    0,
    0,
    x.angle(&S),
```

```
    g[0],
    g[1]));

    loop {
        let (lambda, search_func_calls) =
            minimize(&|lambda: f64| -> f64 { f(&(&x + lambda * &S)) }, 0., eps);
        let dx = &(lambda * &S);
        x += dx;
        iter += 1;
        func_calls += search_func_calls;

        let _g = g;
        g = df(&x);
        func_calls += 1;

        if iter % D::dim() as i32 == 0 {
            S = g.clone();
        } else {
            S = g.clone() + (g.norm() / _g.norm()).powi(2) * S;
        }

        result.push_str(&for-
mat!("\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"
{}\";\"{}\";\n",
        iter,
        x[0],
        x[1],
        if rev { 1./ f(&x) } else {f(&x)},
        S[0],
        S[1],
        lambda,
        dx[0].abs(),
        dx[1].abs(),
        (f(&x) - f(&(&x - dx))).abs(),
        x.angle(&S),
        g[0],
        g[1]));

        if S.norm() < eps {
            return (x, func_calls, iter, result);
        }
    }
}
```

variable_metric_methods.rs

```
use nalgebra::allocator::Allocator;
use nalgebra::DefaultAllocator;
use nalgebra::DimName;
use nalgebra::MatrixN;
use nalgebra::VectorN;
```

```
use super::one_dimension_searchers::minimize;

pub fn broyden<D>(
    f: &dyn Fn(&VectorN<f64, D>) -> f64,
    df: &dyn Fn(&VectorN<f64, D>) -> VectorN<f64, D>,
    mut x: VectorN<f64, D>,
    eps: f64,
    rev: bool,
) -> (VectorN<f64, D>, i32, i32, String)
where
    D: DimName,
    DefaultAllocator: Allocator<f64, D> + Allocator<f64, D, D> + Alloca-
tor<f64, nalgebra::U1, D>,
{
    let mut iter = 0;
    let mut func_calls = 1;
    let mut result = String::new();

    let precision = -eps.log10().round() as usize;
    result.push_str(&for-
mat!("\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"
{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\n",
        "i", "xi", "yi", "f(x, y)", "si1", "si2", "lambdai", "|yi-y(i-1)|", "|yi-y(i-
1)|", "|fi-f(i-1)|",
"angle((xi, yi), si)", "gi1", "gi2", "etai11", "etai12", "etai21", "etai22"));

    let mut g = df(&x);
    let mut eta = MatrixN::<f64, D>::from_diagonal_element(1.);

    result.push_str(&for-
mat!("\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"
{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\n",
    iter,
    x[0],
    x[1],
    if rev { 1./ f(&x) } else {f(&x)},
    (&eta * &g)[0],
    (&eta * &g)[1],
    0,
    0,
    0,
    0,
    x.angle(&(&eta * &g)),
    g[0],
    g[1],
    eta[(0, 0)],
    eta[(0, 1)],
    eta[(1, 0)],
    eta[(1, 1)]));
```

```
    loop {
        #[allow(non_snake_case)]
        let S = &eta * &g;
        let (lambda, search_func_calls) =
            minimize(&|lambda: f64| -> f64 { f(&(&x + lambda * &S)) }, 0., eps);
        let dx = &(lambda * &S);
        x += dx;
        iter += 1;
        func_calls += search_func_calls;
        let dg = &(df(&x) - &g);
        g += dg;
        func_calls += 1;

        if iter % D::dim() as i32 == 0 {
            eta = MatrixN::<f64, D>::from_diagonal_element(1.);
        } else {
            let dx_eta_dg = &(dx - &eta * dg);
            eta += dx_eta_dg * dx_eta_dg.transpose() / dx_eta_dg.dot(dg);
        }

        result.push_str(&for-
mat!("\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"
{}\";\"{}\";\"{}\";\"{}\";\"{}\";\"{}\";\n",
        iter,
        x[0],
        x[1],
        if rev { 1./ f(&x) } else {f(&x)},
        S[0],
        S[1],
        lambda,
        dx[0].abs(),
        dx[1].abs(),
        (f(&x) - f(&(&x - dx))).abs(),
        x.angle(&S),
        g[0],
        g[1],
        eta[(0, 0)],
        eta[(0, 1)],
        eta[(1, 0)],
        eta[(1, 1)]));

        if g.norm() < eps {
            return (x, func_calls, iter, result);
        }
    }
}
```

6. Выводы

В случае квадратичной функции разница между методами сопряжённых градиентов и переменной метрики не сильно различна, а выбор начального приближения слабо влияет на сходимость. Сходимость быстрая, но из-за решения задачи минимизации, количество вызовов функции достаточно велико. На сходимость может сильно повлиять точность: например, при использовании сопряжённых градиентов в модификации Флетчера-Ривса с точностью 10^{-5} было больше итераций, чем при других точностях. Причиной этого может быть неоптимальное решение задачи одномерной минимизации при заданной точности.

В случае функции Розенброка сходимость гораздо хуже, а выбор начально точки гораздо из-за наличия нескольких точек локального экстремума. При использовании сопряжённых градиентов в модификации Флетчера-Ривса точность сильно влияет на сходимость: бо́льшая обеспечивает более оптимальное решение задачи одномерной минимизации и, как следствие, меньшее количество итераций. Метод Бройдена показал более хорошую сходимость.

В случае функции из варианта методы показали почти одинаковую, хорошую сходимость при любой точности. Оба метода нашли одинаковый экстремум, но при этом выбор начального приближения оказался критичным: для разных начальных приближений были получены разные результаты. Это связано с наличием нескольких экстремумов у функции.