Министерство науки и высшего образования Российской Федерации

Новосибирский государственный технический университет

Методы оптимизации

Лабораторная работа №1

Факультет: прикладной математики и информатики

Группа: ПМ-63

Студенты: Кожекин М.В.

Утюганов Д.С.

Преподаватели: Чимитова Е. В.

Новосибирск

2019

1. Цель работы

Ознакомиться с методами одномерного поиска, использу­емыми в многомерных методах минимизации функций n переменных. Сравнить различные алгоритмы по эффективности на тестовых примерах.

1. Задание

Реализовать методы дихотомии, золотого сечения и метод Фибоначчи исследовать их сходимость и провести сравнение по числу вычислений функции для достижения заданной точности ε от 10-1 до 10-7. Построить график зависимости количества вычис­лений минимизируемой функции от десятичного логарифма задаваемой точности ε.

Реализовать алгоритм поиска интервала, содержащего минимум.

1. Исследования

Исследование сходимости методов:

**Метод дихотомии**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 1 | -2.0000000000000000 | 20.0000000000000000 | 22.0000000000000000 | 1.9999999954545453 | 8.9999999749999997 | 9.0000000250000003 | 48.9999996499999995 | 49.0000003500000076 |
| 2 | -2.0000000000000000 | 9.0000000250000003 | 11.0000000250000003 | 1.9999999909090911 | 3.4999999875000003 | 3.5000000375000000 | 2.2499999625000009 | 2.2500001125000013 |
| 3 | -2.0000000000000000 | 3.5000000375000000 | 5.5000000374999996 | 1.9999999818181819 | 0.7499999937500000 | 0.7500000437500000 | 1.5625000156250002 | 1.5624998906250018 |
| 4 | 0.7499999937500000 | 3.5000000375000000 | 2.7500000437500001 | 1.9999999636363650 | 2.1249999906250001 | 2.1250000406249998 | 0.0156249976562501 | 0.0156250101562516 |
| 5 | 0.7499999937500000 | 2.1250000406249998 | 1.3750000468749999 | 1.9999999272727322 | 1.4374999921874998 | 1.4375000421874999 | 0.3164062587890628 | 0.3164062025390644 |
| 6 | 1.4374999921874998 | 2.1250000406249998 | 0.6875000484375000 | 1.9999998545454742 | 1.7812499914062496 | 1.7812500414062498 | 0.0478515662597659 | 0.0478515443847674 |
| 7 | 1.7812499914062496 | 2.1250000406249998 | 0.3437500492187502 | 1.9999997090909936 | 1.9531249910156248 | 1.9531250410156249 | 0.0021972664672853 | 0.0021972617797868 |
| 8 | 1.9531249910156248 | 2.1250000406249998 | 0.1718750496093751 | 1.9999994181821614 | 2.0390624908203123 | 2.0390625408203120 | 0.0015258781890870 | 0.0015258820953385 |
| 9 | 1.9531249910156248 | 2.0390625408203120 | 0.0859375498046873 | 1.9999988363649794 | 1.9960937409179682 | 1.9960937909179683 | 0.0000152588600160 | 0.0000152584693925 |
| 10 | 1.9960937409179682 | 2.0390625408203120 | 0.0429687999023438 | 1.9999976727326874 | 2.0175781158691404 | 2.0175781658691401 | 0.0003089901575089 | 0.0003089919153230 |
| 11 | 1.9960937409179682 | 2.0175781658691401 | 0.0214844249511719 | 1.9999953454762485 | 2.0068359283935542 | 2.0068359783935539 | 0.0000467299170018 | 0.0000467306005971 |
| 12 | 1.9960937409179682 | 2.0068359783935539 | 0.0107422374755857 | 1.9999906909957434 | 2.0014648346557613 | 2.0014648846557610 | 0.0000021457405687 | 0.0000021458870547 |
| 13 | 1.9960937409179682 | 2.0014648846557610 | 0.0053711437377928 | 1.9999813821648003 | 1.9987792877868646 | 1.9987793377868648 | 0.0000014901383073 | 0.0000014900162386 |
| 14 | 1.9987792877868646 | 2.0014648846557610 | 0.0026855968688964 | 1.9999627650228351 | 2.0001220612213131 | 2.0001221112213128 | 0.0000000148989417 | 0.0000000149111504 |
| 15 | 1.9987792877868646 | 2.0001221112213128 | 0.0013428234344481 | 1.9999255328184540 | 1.9994506745040888 | 1.9994507245040889 | 0.0000003017585005 | 0.0000003017035704 |
| 16 | 1.9994506745040888 | 2.0001221112213128 | 0.0006714367172240 | 1.9998510767268045 | 1.9997863678627008 | 1.9997864178627009 | 0.0000000456386901 | 0.0000000456173294 |
| 17 | 1.9997863678627008 | 2.0001221112213128 | 0.0003357433586120 | 1.9997021978032867 | 1.9999542145420068 | 1.9999542645420070 | 0.0000000020963082 | 0.0000000020917321 |
| 18 | 1.9999542145420068 | 2.0001221112213128 | 0.0001678966793059 | 1.9994045729313510 | 2.0000381378816598 | 2.0000381878816595 | 0.0000000014544980 | 0.0000000014583143 |
| 19 | 1.9999542145420068 | 2.0000381878816595 | 0.0000839733396527 | 1.9988098544969795 | 1.9999961762118332 | 1.9999962262118334 | 0.0000000000146214 | 0.0000000000142415 |
| 20 | 1.9999961762118332 | 2.0000381878816595 | 0.0000420116698263 | 1.9976225385401400 | 2.0000171570467464 | 2.0000172070467461 | 0.0000000002943643 | 0.0000000002960825 |
| 21 | 1.9999961762118332 | 2.0000172070467461 | 0.0000210308349129 | 1.9952563548716202 | 2.0000066666292899 | 2.0000067166292896 | 0.0000000000444439 | 0.0000000000451131 |
| 22 | 1.9999961762118332 | 2.0000067166292896 | 0.0000105404174564 | 1.9905575016047525 | 2.0000014214205617 | 2.0000014714205614 | 0.0000000000020204 | 0.0000000000021651 |
| 23 | 1.9999961762118332 | 2.0000014714205614 | 0.0000052952087282 | 1.9812916565463803 | 1.9999987988161971 | 1.9999988488161973 | 0.0000000000014428 | 0.0000000000013252 |
| 24 | 1.9999987988161971 | 2.0000014714205614 | 0.0000026726043643 | 1.9632704625467756 | 2.0000001101183793 | 2.0000001601183790 | 0.0000000000000121 | 0.0000000000000256 |
| 25 | 1.9999987988161971 | 2.0000001601183790 | 0.0000013613021819 | 1.9291434520660331 | 1.9999994544672879 | 1.9999995044672880 | 0.0000000000002976 | 0.0000000000002456 |
| 26 | 1.9999994544672879 | 2.0000001601183790 | 0.0000007056510911 | 1.8676637910639433 | 1.9999997822928333 | 1.9999998322928334 | 0.0000000000000474 | 0.0000000000000281 |
| 27 | 1.9999997822928333 | 2.0000001601183790 | 0.0000003778255457 | 1.7662598654150610 | 1.9999999462056059 | 1.9999999962056061 | 0.0000000000000029 | 0.0000000000000000 |
| 28 | 1.9999999462056059 | 2.0000001601183790 | 0.0000002139127731 | 1.6210869268983246 | 2.0000000281619927 | 2.0000000781619924 | 0.0000000000000008 | 0.0000000000000061 |
| 29 | 1.9999999462056059 | 2.0000000781619924 | 0.0000001319563865 | 1.4504177522236954 | 1.9999999871837992 | 2.0000000371837992 | 0.0000000000000002 | 0.0000000000000014 |

**Метод золотого сечения**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 1 | -2.0000000000000000 | 20.0000000000000000 | 22.0000000000000000 | 1.6180339887498947 | 3.1934955049953730 | 6.4032522475023121 | 1.4244315204441604 | 19.3886303551341648 |
| 2 | -2.0000000000000000 | 11.5967477524976879 | 13.5967477524976879 | 1.6180339887498951 | 1.2097567425069391 | 3.1934955049953730 | 0.6244844060132442 | 1.4244315204441604 |
| 3 | -2.0000000000000000 | 6.4032522475023121 | 8.4032522475023121 | 1.6180339887498947 | -0.0162612375115672 | 1.2097567425069391 | 4.0653093778916753 | 0.6244844060132442 |
| 4 | -2.0000000000000000 | 3.1934955049953730 | 5.1934955049953730 | 1.6180339887498947 | 1.2097567425069391 | 1.9674775249768663 | 0.6244844060132442 | 0.0010577113816304 |
| 5 | -0.0162612375115672 | 3.1934955049953730 | 3.2097567425069400 | 1.6180339887498942 | 1.9674775249768663 | 2.4357747225254460 | 0.0010577113816304 | 0.1898996087921295 |
| 6 | 1.2097567425069391 | 3.1934955049953730 | 1.9837387624884339 | 1.6180339887498951 | 1.6780539400555192 | 1.9674775249768663 | 0.1036492655137752 | 0.0010577113816304 |
| 7 | 1.2097567425069391 | 2.4357747225254460 | 1.2260179800185069 | 1.6180339887498947 | 1.9674775249768663 | 2.1463511376040993 | 0.0010577113816304 | 0.0214186554780140 |
| 8 | 1.6780539400555192 | 2.4357747225254460 | 0.7577207824699268 | 1.6180339887498953 | 1.8569275526827529 | 1.9674775249768663 | 0.0204697251813464 | 0.0010577113816304 |
| 9 | 1.6780539400555192 | 2.1463511376040993 | 0.4682971975485801 | 1.6180339887498949 | 1.9674775249768663 | 2.0358011653099868 | 0.0010577113816304 | 0.0012817234375530 |
| 10 | 1.8569275526827529 | 2.1463511376040993 | 0.2894235849213465 | 1.6180339887498929 | 1.9252511930158738 | 1.9674775249768663 | 0.0055873841455502 | 0.0010577113816304 |
| 11 | 1.8569275526827529 | 2.0358011653099868 | 0.1788736126272339 | 1.6180339887498936 | 1.9674775249768663 | 1.9935748333489949 | 0.0010577113816304 | 0.0000412827664932 |
| 12 | 1.9252511930158738 | 2.0358011653099868 | 0.1105499722941130 | 1.6180339887499087 | 1.9935748333489949 | 2.0097038569378580 | 0.0000412827664932 | 0.0000941648394704 |
| 13 | 1.9674775249768663 | 2.0358011653099868 | 0.0683236403331204 | 1.6180339887498925 | 1.9836065485657293 | 1.9935748333489949 | 0.0002687452499278 | 0.0000412827664932 |
| 14 | 1.9674775249768663 | 2.0097038569378580 | 0.0422263319609917 | 1.6180339887499011 | 1.9935748333489949 | 1.9997355721545920 | 0.0000412827664932 | 0.0000000699220854 |
| 15 | 1.9836065485657293 | 2.0097038569378580 | 0.0260973083721288 | 1.6180339887498563 | 1.9997355721545920 | 2.0035431181322609 | 0.0000000699220854 | 0.0000125536860992 |
| 16 | 1.9935748333489949 | 2.0097038569378580 | 0.0161290235888631 | 1.6180339887499235 | 1.9973823793266641 | 1.9997355721545920 | 0.0000068519379895 | 0.0000000699220854 |
| 17 | 1.9935748333489949 | 2.0035431181322609 | 0.0099682847832661 | 1.6180339887498782 | 1.9997355721545920 | 2.0011899253043333 | 0.0000000699220854 | 0.0000014159222299 |
| 18 | 1.9973823793266641 | 2.0035431181322609 | 0.0061607388055969 | 1.6180339887499386 | 1.9988367324764058 | 1.9997355721545920 | 0.0000013531913314 | 0.0000000699220854 |
| 19 | 1.9973823793266641 | 2.0011899253043333 | 0.0038075459776692 | 1.6180339887499331 | 1.9997355721545920 | 2.0002910856261473 | 0.0000000699220854 | 0.0000000847308417 |
| 20 | 1.9988367324764058 | 2.0011899253043333 | 0.0023531928279275 | 1.6180339887500421 | 1.9993922459479616 | 1.9997355721545920 | 0.0000003693649878 | 0.0000000699220854 |
| 21 | 1.9988367324764058 | 2.0002910856261473 | 0.0014543531497415 | 1.6180339887499093 | 1.9997355721545920 | 1.9999477594195174 | 0.0000000699220854 | 0.0000000027290782 |
| 22 | 1.9993922459479616 | 2.0002910856261473 | 0.0008988396781857 | 1.6180339887511506 | 1.9999477594195174 | 2.0000788983612217 | 0.0000000027290782 | 0.0000000062249514 |
| 23 | 1.9997355721545920 | 2.0002910856261473 | 0.0005555134715554 | 1.6180339887497466 | 1.9998667110962960 | 1.9999477594195174 | 0.0000000177659319 | 0.0000000027290782 |
| 24 | 1.9997355721545920 | 2.0000788983612217 | 0.0003433262066297 | 1.6180339887502830 | 1.9999477594195174 | 1.9999978500380000 | 0.0000000027290782 | 0.0000000000046223 |
| 25 | 1.9998667110962960 | 2.0000788983612217 | 0.0002121872649257 | 1.6180339887461390 | 1.9999978500380000 | 2.0000288077427393 | 0.0000000000046223 | 0.0000000008298860 |
| 26 | 1.9999477594195174 | 2.0000788983612217 | 0.0001311389417042 | 1.6180339887508619 | 1.9999787171242569 | 1.9999978500380000 | 0.0000000004529608 | 0.0000000000046223 |
| 27 | 1.9999477594195174 | 2.0000288077427393 | 0.0000810483232219 | 1.6180339887473629 | 1.9999978500380000 | 2.0000096748289966 | 0.0000000000046223 | 0.0000000000936023 |
| 28 | 1.9999787171242569 | 2.0000288077427393 | 0.0000500906184824 | 1.6180339887449178 | 1.9999905419152537 | 1.9999978500380000 | 0.0000000000894554 | 0.0000000000046223 |
| 29 | 1.9999787171242569 | 2.0000096748289966 | 0.0000309577047397 | 1.6180339887441468 | 1.9999978500380000 | 2.0000023667062505 | 0.0000000000046223 | 0.0000000000056013 |
| 30 | 1.9999905419152537 | 2.0000096748289966 | 0.0000191329137429 | 1.6180339887649431 | 1.9999950585835047 | 1.9999978500380000 | 0.0000000000244176 | 0.0000000000046223 |
| 31 | 1.9999905419152537 | 2.0000023667062505 | 0.0000118247909968 | 1.6180339887596593 | 1.9999978500380000 | 1.9999995752517556 | 0.0000000000046223 | 0.0000000000001804 |
| 32 | 1.9999950585835047 | 2.0000023667062505 | 0.0000073081227459 | 1.6180339888834203 | 1.9999995752517556 | 2.0000006414924947 | 0.0000000000001804 | 0.0000000000004115 |
| 33 | 1.9999978500380000 | 2.0000023667062505 | 0.0000045166682505 | 1.6180339887864374 | 1.9999989162787388 | 1.9999995752517556 | 0.0000000000011745 | 0.0000000000001804 |
| 34 | 1.9999978500380000 | 2.0000006414924947 | 0.0000027914544947 | 1.6180339886542248 | 1.9999995752517556 | 1.9999999825194779 | 0.0000000000001804 | 0.0000000000000003 |
| 35 | 1.9999989162787388 | 2.0000006414924947 | 0.0000017252137559 | 1.6180339886634068 | 1.9999999825194779 | 2.0000002342247725 | 0.0000000000000003 | 0.0000000000000549 |
| 36 | 1.9999995752517556 | 2.0000006414924947 | 0.0000010662407390 | 1.6180339889763238 | 1.9999998269570505 | 1.9999999825194779 | 0.0000000000000299 | 0.0000000000000003 |
| 37 | 1.9999995752517556 | 2.0000002342247725 | 0.0000006589730168 | 1.6180339890392572 | 1.9999999825194779 | 2.0000000786623451 | 0.0000000000000003 | 0.0000000000000062 |
| 38 | 1.9999998269570505 | 2.0000002342247725 | 0.0000004072677220 | 1.6180339894197009 | 1.9999999230999179 | 1.9999999825194779 | 0.0000000000000059 | 0.0000000000000003 |
| 39 | 1.9999998269570505 | 2.0000000786623451 | 0.0000002517052946 | 1.6180339893058473 | 1.9999999825194779 | 2.0000000192427851 | 0.0000000000000003 | 0.0000000000000004 |
| 40 | 1.9999999230999179 | 2.0000000786623451 | 0.0000001555624272 | 1.6180339910312869 | 1.9999999598232254 | 1.9999999825194779 | 0.0000000000000016 | 0.0000000000000003 |

**Метод Фибоначчи**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 1 | -2.0000000000000000 | 20.0000000000000000 | 22.0000000000000000 | 1.6180339887498949 | 6.4032522475023139 | 6.4032522475023139 | 19.3886303551341790 | 19.3886303551341790 |
| 2 | -2.0000000000000000 | 11.5967477524976861 | 13.5967477524976861 | 1.6180339887498947 | 3.1934955049953739 | 6.4032522475023139 | 1.4244315204441627 | 19.3886303551341790 |
| 3 | -2.0000000000000000 | 6.4032522475023139 | 8.4032522475023139 | 1.0000000000000000 | 1.2097567425069395 | 3.1934955049953739 | 0.6244844060132435 | 1.4244315204441627 |
| 4 | -2.0000000000000000 | 6.4032522475023139 | 8.4032522475023139 | 1.6180339887498947 | -0.0162612375115661 | 1.2097567425069395 | 4.0653093778916718 | 0.6244844060132435 |
| 5 | -2.0000000000000000 | 3.1934955049953739 | 5.1934955049953739 | 1.6180339887498951 | 1.2097567425069395 | 1.9674775249768679 | 0.6244844060132435 | 0.0010577113816303 |
| 6 | -0.0162612375115661 | 3.1934955049953739 | 3.2097567425069400 | 1.6180339887498940 | 1.9674775249768679 | 2.4357747225254451 | 0.0010577113816303 | 0.1898996087921287 |
| 7 | 1.2097567425069395 | 3.1934955049953739 | 1.9837387624884344 | 1.6180339887498971 | 1.6780539400555168 | 1.9674775249768679 | 0.1036492655137768 | 0.0010577113816303 |
| 8 | 1.2097567425069395 | 2.4357747225254451 | 1.2260179800185056 | 1.6180339887498896 | 1.9674775249768679 | 2.1463511376040940 | 0.0010577113816303 | 0.0214186554780125 |
| 9 | 1.6780539400555168 | 2.4357747225254451 | 0.7577207824699284 | 1.6180339887499087 | 1.8569275526827429 | 1.9674775249768679 | 0.0204697251813493 | 0.0010577113816303 |
| 10 | 1.6780539400555168 | 2.1463511376040940 | 0.4682971975485772 | 1.6180339887498587 | 1.9674775249768679 | 2.0358011653099690 | 0.0010577113816303 | 0.0012817234375517 |
| 11 | 1.8569275526827429 | 2.1463511376040940 | 0.2894235849213511 | 1.6180339887499893 | 1.9252511930158440 | 1.9674775249768679 | 0.0055873841455546 | 0.0010577113816303 |
| 12 | 1.8569275526827429 | 2.0358011653099690 | 0.1788736126272261 | 1.6180339887496478 | 1.9674775249768679 | 1.9935748333489451 | 0.0010577113816303 | 0.0000412827664938 |
| 13 | 1.9252511930158440 | 2.0358011653099690 | 0.1105499722941250 | 1.6180339887505417 | 1.9935748333489451 | 2.0097038569378918 | 0.0000412827664938 | 0.0000941648394711 |
| 14 | 1.9674775249768679 | 2.0358011653099690 | 0.0683236403331011 | 1.6180339887482014 | 1.9836065485658143 | 1.9935748333489451 | 0.0002687452499250 | 0.0000412827664938 |
| 15 | 1.9674775249768679 | 2.0097038569378918 | 0.0422263319610239 | 1.6180339887543149 | 1.9935748333489451 | 1.9997355721547607 | 0.0000412827664938 | 0.0000000699220853 |
| 16 | 1.9836065485658143 | 2.0097038569378918 | 0.0260973083720775 | 1.6180339887383008 | 1.9997355721547607 | 2.0035431181320758 | 0.0000000699220853 | 0.0000125536860978 |
| 17 | 1.9935748333489451 | 2.0097038569378918 | 0.0161290235889466 | 1.6180339887802846 | 1.9973823793262602 | 1.9997355721547607 | 0.0000068519379916 | 0.0000000699220853 |
| 18 | 1.9935748333489451 | 2.0035431181320758 | 0.0099682847831306 | 1.6180339886704505 | 1.9997355721547607 | 2.0011899253035752 | 0.0000000699220853 | 0.0000014159222281 |
| 19 | 1.9973823793262602 | 2.0035431181320758 | 0.0061607388058156 | 1.6180339889578832 | 1.9988367324750747 | 1.9997355721547607 | 0.0000013531913345 | 0.0000000699220853 |
| 20 | 1.9973823793262602 | 2.0011899253035752 | 0.0038075459773150 | 1.6180339882053743 | 1.9997355721547607 | 2.0002910856238891 | 0.0000000699220853 | 0.0000000847308404 |
| 21 | 1.9988367324750747 | 2.0011899253035752 | 0.0023531928285005 | 1.6180339901754683 | 1.9993922459442028 | 1.9997355721547607 | 0.0000003693649923 | 0.0000000699220853 |
| 22 | 1.9988367324750747 | 2.0002910856238891 | 0.0014543531488145 | 1.6180339850172953 | 1.9997355721547607 | 1.9999477594133310 | 0.0000000699220853 | 0.0000000027290789 |
| 23 | 1.9993922459442028 | 2.0002910856238891 | 0.0008988396796863 | 1.6180339985213206 | 1.9999477594133310 | 2.0000788983653188 | 0.0000000027290789 | 0.0000000062249520 |
| 24 | 1.9997355721547607 | 2.0002910856238891 | 0.0005555134691284 | 1.6180339631669240 | 1.9998667111067485 | 1.9999477594133310 | 0.0000000177659291 | 0.0000000027290789 |
| 25 | 1.9997355721547607 | 2.0000788983653188 | 0.0003433262105581 | 1.6180340557269848 | 1.9999477594133310 | 1.9999978500587363 | 0.0000000027290789 | 0.0000000000046222 |
| 26 | 1.9998667111067485 | 2.0000788983653188 | 0.0002121872585703 | 1.6180338134016159 | 1.9999978500587363 | 2.0000288077199135 | 0.0000000000046222 | 0.0000000008298847 |
| 27 | 1.9999477594133310 | 2.0000788983653188 | 0.0001311389519878 | 1.6180344478177793 | 1.9999787170745078 | 1.9999978500587363 | 0.0000000004529629 | 0.0000000000046222 |
| 28 | 1.9999477594133310 | 2.0000288077199135 | 0.0000810483065825 | 1.6180327868811182 | 1.9999978500587363 | 2.0000096747356846 | 0.0000000000046222 | 0.0000000000936005 |
| 29 | 1.9999787170745078 | 2.0000288077199135 | 0.0000500906454057 | 1.6180371352893210 | 1.9999905417514561 | 1.9999978500587363 | 0.0000000000894585 | 0.0000000000046222 |
| 30 | 1.9999787170745078 | 2.0000096747356846 | 0.0000309576611768 | 1.6180257510822258 | 1.9999978500587363 | 2.0000023664284043 | 0.0000000000046222 | 0.0000000000056000 |
| 31 | 1.9999905417514561 | 2.0000096747356846 | 0.0000191329842285 | 1.6180555555313005 | 1.9999950581211239 | 1.9999978500587363 | 0.0000000000244222 | 0.0000000000046222 |
| 32 | 1.9999905417514561 | 2.0000023664284043 | 0.0000118246769483 | 1.6179775281042255 | 1.9999978500587363 | 1.9999995744907917 | 0.0000000000046222 | 0.0000000000001811 |
| 33 | 1.9999950581211239 | 2.0000023664284043 | 0.0000073083072805 | 1.6181818180647176 | 1.9999995744907917 | 2.0000006419963490 | 0.0000000000001811 | 0.0000000000004122 |
| 34 | 1.9999978500587363 | 2.0000023664284043 | 0.0000045163696680 | 1.6176470590013039 | 1.9999989175642940 | 1.9999995744907917 | 0.0000000000011717 | 0.0000000000001811 |
| 35 | 1.9999978500587363 | 2.0000006419963490 | 0.0000027919376127 | 1.6190476189985661 | 1.9999995744907917 | 1.9999999850698518 | 0.0000000000001811 | 0.0000000000000002 |
| 36 | 1.9999989175642940 | 2.0000006419963490 | 0.0000017244320549 | 1.6153846161846279 | 1.9999999850698518 | 2.0000002314172889 | 0.0000000000000002 | 0.0000000000000536 |
| 37 | 1.9999995744907917 | 2.0000006419963490 | 0.0000010675055573 | 1.6249999989859842 | 1.9999998208382281 | 1.9999999850698518 | 0.0000000000000321 | 0.0000000000000002 |
| 38 | 1.9999995744907917 | 2.0000002314172889 | 0.0000006569264972 | 1.6000000000000001 | 1.9999999850698518 | 2.0000000671856646 | 0.0000000000000002 | 0.0000000000000045 |
| 39 | 1.9999998208382281 | 2.0000002314172889 | 0.0000004105790608 | 1.6666666666666667 | 1.9999999029540403 | 1.9999999850698518 | 0.0000000000000094 | 0.0000000000000002 |
| 40 | 1.9999998208382281 | 2.0000000671856646 | 0.0000002463474365 | 1.5000000000000000 | 1.9999999850698518 | 1.9999999850698524 | 0.0000000000000002 | 0.0000000000000002 |

**График зависимости количества вычис­лений целевой функции от логарифма задаваемой точности ε:**

**Поиск интервала, содержащего минимум**

|  |  |  |  |
| --- | --- | --- | --- |
| i | Xi | f(xi) | Интервал, содержащий минимум |
| 1 | 1.9900000 | 0.0001000 | [ 1.9900000 ; 1.9900001 ] |
| 2 | 1.9900001 | 0.0001000 | [ 1.9900000 ; 1.9900003 ] |
| 3 | 1.9900003 | 0.0001000 | [ 1.9900001 ; 1.9900007 ] |
| 4 | 1.9900007 | 0.0001000 | [ 1.9900003 ; 1.9900015 ] |
| 5 | 1.9900015 | 0.0001000 | [ 1.9900007 ; 1.9900031 ] |
| 6 | 1.9900031 | 0.0000999 | [ 1.9900015 ; 1.9900064 ] |
| 7 | 1.9900064 | 0.0000999 | [ 1.9900031 ; 1.9900127 ] |
| 8 | 1.9900127 | 0.0000997 | [ 1.9900064 ; 1.9900255 ] |
| 9 | 1.9900255 | 0.0000995 | [ 1.9900127 ; 1.9900511 ] |
| 10 | 1.9900511 | 0.0000990 | [ 1.9900255 ; 1.9901023 ] |
| 11 | 1.9901023 | 0.0000980 | [ 1.9900511 ; 1.9902047 ] |
| 12 | 1.9902047 | 0.0000959 | [ 1.9901023 ; 1.9904095 ] |
| 13 | 1.9904095 | 0.0000920 | [ 1.9902047 ; 1.9908191 ] |
| 14 | 1.9908191 | 0.0000843 | [ 1.9904095 ; 1.9916383 ] |
| 15 | 1.9916383 | 0.0000699 | [ 1.9908191 ; 1.9932767 ] |
| 16 | 1.9932767 | 0.0000452 | [ 1.9916383 ; 1.9965536 ] |
| 17 | 1.9965536 | 0.0000119 | [ 1.9932767 ; 2.0031071 ] |
| 18 | 2.0031071 | 0.0000097 | [ 1.9965536 ; 2.0162143 ] |

1. Исходный код

**Текст программы**

|  |
| --- |
| #include <iostream>  #include <iomanip>  #include <fstream>  #include <cmath>  using namespace std;  typedef double real;  /\* Заданная функция  \* x - аргумент функции  \*/  real f(real x) {  return (x - 2)\*(x - 2);  }  /\* Вычисление n-го числа Фибоначчи  \* n - номер числа  \*/  int F(real n) {  int tmp, prev = 1, current = 1;  if (n == 1 || n == 2)  return 1;  for (int i = 0; i < n - 2; ++i) {  tmp = current;  current += prev;  prev = tmp;  }  return current;  }  /\* Метод дихотомии  \* a,b - границы отрезка  \* E - точность  \* fileName - файл, в который будет сохранён результат  \* return количество вычислений целевой функции  \*/  int calcByDichotomy(real a, real b, real E, const char \*fileName) {  ofstream fout(fileName);  real aPrev, bPrev, x1, x2, f1, f2, delta = E / 2;  int k = 1;  fout << setprecision(numeric\_limits<real>::digits10 + 1) << fixed;  fout << "k \t a \t b \t (b - a) \t (bPrev - aPrev) / (b - a) \t x1 \t x2 \t f1 \t f2 " << endl;  while (abs(b - a) >= E) {  fout << k << "\t" << a << "\t" << b << "\t" << b - a << "\t";  aPrev = a;  bPrev = b;  x1 = (a + b - delta) / 2;  x2 = (a + b + delta) / 2;  f1 = f(x1);  f2 = f(x2);  if (f1 > f2) { // [x1, b]  a = x1;  }  else { // [a, x2]  b = x2;  }  k++;  fout << (bPrev - aPrev) / (b - a) << "\t" << x1 << "\t" << x2 << "\t" << f1 << "\t" << f2 << endl;  }  fout.close();  return k \* 2;  }  /\* Метод золотого сечения  \* a,b - границы отрезка  \* E - точность  \* fileName - файл, в который будет сохранён результат  \* return количество вычислений целевой функции  \*/  int calcByGoldenSectionSearch(real a, real b, real E, const char \*fileName) {  ofstream fout(fileName);  real aPrev, bPrev, x1, x2, f1, f2, delta = E / 2;  real alpha = (3 - sqrt(5)) / 2;  real beta = (sqrt(5) - 1) / 2;  int k = 1;  fout << setprecision(numeric\_limits<real>::digits10 + 1) << fixed;  x1 = a + alpha \* (b - a);  x2 = a + beta \* (b - a);  f1 = f(x1);  f2 = f(x2);  fout << "k \t a \t b \t (b - a) \t (bPrev - aPrev) / (b - a) \t x1 \t x2 \t f1 \t f2 " << endl;  while (abs(b - a) >= E) {  fout << k << "\t" << a << "\t" << b << "\t" << b - a << "\t";  aPrev = a;  bPrev = b;  if (f1 > f2) { // [x1, b]  a = x1;  x1 = x2;  f1 = f2;  x2 = a + beta \* (b - a);  f2 = f(x2);  }  else { // [a, x2]  b = x2;  x2 = x1;  f2 = f1;  x1 = a + alpha \* (b - a);  f1 = f(x1);  }  k++;  fout << (bPrev - aPrev) / (b - a) << "\t" << x1 << "\t" << x2 << "\t" << f1 << "\t" << f2 << endl;  }  fout.close();  return k + 1;  }  /\* Метод Фибоначчи  \* a,b - границы отрезка  \* E - точность  \* fileName - файл, в который будет сохранён результат  \* return количество вычислений целевой функции  \*/  int calcByFibonacci(real a, real b, real E, const char \*fileName) {  ofstream fout(fileName);  real aPrev, bPrev, length = b - a, x1, x2, f1, f2, delta = E / 2;  int k = 1, n = 0;  while (F(n + 2) < (b - a) / E) {  ++n;  }  fout << setprecision(numeric\_limits<real>::digits10 + 1) << fixed;  x1 = a + F(n) \* (b - a) / F(n + 2);  x2 = a + F(n + 1) \* (b - a) / F(n + 2);  f1 = f(x1);  f2 = f(x2);  real Fn2 = F(n + 2);  fout << "k \t a \t b \t (b - a) \t (bPrev - aPrev) / (b - a) \t x1 \t x2 \t f1 \t f2 " << endl;  while (abs(b - a) >= E) {  fout << k << "\t" << a << "\t" << b << "\t" << b - a << "\t";  aPrev = a;  bPrev = b;  if (f1 > f2) { // [x1, b]  a = x1;  x1 = x2;  f1 = f2;  x2 = a + F(n - k + 2) \* length / Fn2;  f2 = f(x2);  }  else { // [a, x2]  b = x2;  x2 = x1;  f2 = f1;  x1 = a + F(n - k + 1) \* length / Fn2;  f1 = f(x1);  }  k++;  fout << (bPrev - aPrev) / (b - a) << "\t" << x1 << "\t" << x2 << "\t" << f1 << "\t" << f2 << endl;  }  fout << "x = " << x1 << "\tx2 - x1 = " << x2 - x1 << endl;  fout.close();  return k + 1;  }  /\* Поиск интервала, содержащего минимум  \* a,b - границы отрезка  \* x0 - начальная точка  \* delta - точность  \* fileName - файл, в который будет сохранён результат  \*/  void findRange(real a, real b, real x0, real delta, const char \*fileName) {  ofstream fout(fileName);  real x = x0, xNext = x0, xPrev = x0, h, fx, fxNext;  int k = 0;  fout << setprecision(7) << fixed;  cout << f(x0) << "\t" << f(x0 + delta) << endl;  cout << f(x0) << "\t" << f(x0 - delta) << endl;  // Шаг 1  if (f(x0) > f(x0 + delta)) {  x = x0;  xNext = x + delta;  h = delta;  }  else if (f(x0) > f(x0 - delta)) {  x = x0;  xNext = x - delta;  h = -delta;  }  else {  fout << ++k << "\t" << x0 << "\t" << f(x0)  << "\t[ " << x0 - delta << " ; " << x0 + delta << " ]" << endl;  return;  }  fx = f(x);  fxNext = f(xNext);  do {  // Сохраняем текущий шаг, как предыдущий  xPrev = x;  fx = fxNext;  x = xNext;  // Выполняем шаг 2  h \*= 2;  xNext = x + h;  fxNext = f(xNext);  fout << ++k << "\t" << x << "\t" << fx  << "\t[ " << xPrev << " ; " << xNext << " ]" << endl;  } while (fx > fxNext);  fout.close();  }  /\* Построение таблицы для графика зависисмости количества вычислений  \* целевой функции от логарифма задаваемой точности E  \* a,b - границы отрезка  \* E0,E1 - границы точности E  \* step - во сколько раз повышается точность на каждой итерации  \* fileName - файл, в который будет сохранён результат  \*/  void createTable(real a, real b, real E0, real E1, real step) {  ofstream fout("Table.txt");  fout << "ln(E)\tD\tGS\tF" << endl;  for (real E = E0; E >= E1; E /= step) {  fout << log(E) << "\t"  << calcByDichotomy(a, b, E, "Dichotomy.txt") << "\t"  << calcByGoldenSectionSearch(a, b, E, "GoldenSection.txt") << "\t"  << calcByFibonacci(a, b, E, "Fibonacci.txt") << endl;  }  fout.close();  }  void main() {  real a = -2, b = 20, E = 1e-2, x0 = -20;  /\*calcByDichotomy(a, b, E, "Dichotomy.txt");  calcByGoldenSectionSearch(a, b, E, "GoldenSection.txt");  calcByFibonacci(a, b, E, "Fibonacci.txt");\*/  findRange(a, b, x0, 1e-1, "Range.txt");  //createTable(a, b, 1e-1, 1e-7, 2);  } |

1. Выводы

Методы Фибоначчи и золотого сечения эффективней, чем метод дихотомии, т.к. требуют в 2 раза меньше вычислений целевой функции.

Также стоит отметить, что на практике методы золотого сечения и Фибоначчи имеют схожую эффективность. Однако метод Фибоначчи не позволяет определить минимальный интервал на последней итерации.