|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Министерство науки и высшего образования РФ | | | | | | | | |  | |
|  | | | | |  | | |  | | | | |
|  | | | ФГБОУ ВО «Пермский государственный национальный исследовательский университет» | | | | | | |  | | |
|  | | | | |  | | |  | | | | |
|  | | | | **Численные методы**  Лабораторная работа №3  Варианты: №21b, №25b | | | | |  | | | |
|  | | | | |  | | |  | | | | |
|  | Работу выполнили  Студенты гр.ПМИ-4:  Колесников А.С  Пухов Н.А. | | | | |  | Проверил  профессор, доктор физико-математических наук  Русаков С.В  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г. | | | | |  | |
|  | | | | |  | | |  | | | | |
|  | | | | | Пермь 2020 | | |  | | | | |

СОДЕРЖАНИЕ

[Задание 3](#_Toc54715244)

[Исходные данные 4](#_Toc54715245)

[Тестирование 5](#_Toc54715246)

[Краткие выводы 23](#_Toc54715247)

[Текст программы 24](#_Toc54715248)

Задание

1. Решить систему линейных алгебраических уравнений  итерационными методами с критерием остановки  (по невязке):
   * методом простой итерации;
   * градиентным методом наискорейшего спуска;
   * методом ПВР;
   * методом сопряженных градиентов.

В качества начального приближения выбирать вектор , который во всех вариантах принять равным .

1. Для каждого метода получить число итераций, необходимое для достижения требуемой точности (по невязке), выдавая (на печать) на каждом шаге (или через заданное число шагов)

* значение параметров итерационного метода;
* значение нормы невязки;
* оценку нормы матрицы перехода *q*;
* оценку погрешности приближенного решения.

1. Оценку нормы матрицы перехода осуществлять по формуле



1. В методе простой итерации значения итерационного параметра вычислять по формуле



1. В методе ПВР получить решение при оптимальном значении параметра , которое необходимо определить, варьируя параметр в диапазоне (0, 2) с шагом 0.1, и производя вычисления с критерием остановки(или по критерию минимальности нормы вектора невязки при заданном числе итераций).
2. Провести анализ эффективности рассматриваемых методов.
3. Сравнить решение, полученное итерационным методом, с решением полученным прямым методом.
4. Сравнить фактическое число итераций, необходимое для достижения заданной точности, с теоретической оценкой, вычислив число обусловленности в эвклидовой норме.

Исходные данные

Вариант №21(b):



Вариант №25(b):



Тестирование

Вариант 25b:

A:

5.4000000 2.6000000 -1.2000000 2.1000000

2.6000000 2.9000000 -0.4000000 2.5000000

-1.2000000 -0.4000000 9.8000000 -2.8000000

2.1000000 2.5000000 -2.8000000 5.5000000

vector b:

1.0000000 2.0000000 3.0000000 4.0000000

Норма матрицы 12.461100

Метод простой итерации

| Норма | Оценка |

Itr| Tau | q | невязки |погрешности| x[1] | x[2] | x[3] | x[4] |

1| 0.1444| 0.578| 6.9228332| 0.4222019| 0.14445| 0.28890| 0.43335| 0.57780|

2| 0.1444| 0.306| 1.6920848| 0.4009164| -0.03243| 0.21891| 0.52867| 0.72368|

3| 0.1444| 0.834| 1.1719037| 0.0294135| -0.07278| 0.19750| 0.51335| 0.87115|

4| 0.1444| 0.392| 0.4370527| 0.0896879| -0.12101| 0.14607| 0.57113| 0.91523|

5| 0.1444| 1.136| 0.4632147| -0.0078502| -0.11566| 0.12171| 0.55362| 0.98086|

6| 0.1444| 0.623| 0.2845483| 0.0247562| -0.12828| 0.08084| 0.58696| 0.99444|

7| 0.1444| 0.853| 0.2344987| 0.0060084| -0.11404| 0.05885| 0.57405| 1.02930|

8| 0.1444| 0.902| 0.2104621| 0.0034047| -0.11547| 0.02739| 0.59471| 1.03487|

9| 0.1444| 0.677| 0.1395902| 0.0101650| -0.10207| 0.00883| 0.58631| 1.05616|

10| 0.1444| 1.127| 0.1567351| -0.0026976| -0.10007| -0.01517| 0.59967| 1.05978|

11| 0.1444| 0.635| 0.0981652| 0.0087648| -0.08940| -0.03040| 0.59454| 1.07399|

12| 0.1444| 1.200| 0.1175035| -0.0030528| -0.08653| -0.04868| 0.60338| 1.07709|

13| 0.1444| 0.673| 0.0784134| 0.0059692| -0.07844| -0.06099| 0.60041| 1.08704|

14| 0.1444| 1.134| 0.0886524| -0.0016440| -0.07557| -0.07495| 0.60636| 1.08987|

15| 0.1444| 0.707| 0.0622670| 0.0040895| -0.06953| -0.08481| 0.60472| 1.09703|

16| 0.1444| 1.083| 0.0672658| -0.0008168| -0.06695| -0.09550| 0.60878| 1.09957|

17| 0.1444| 0.736| 0.0492527| 0.0028238| -0.06244| -0.10336| 0.60795| 1.10481|

18| 0.1444| 1.043| 0.0512874| -0.0003411| -0.06022| -0.11156| 0.61074| 1.10702|

19| 0.1444| 0.760| 0.0388490| 0.0019673| -0.05684| -0.11779| 0.61038| 1.11089|

20| 0.1444| 1.012| 0.0392638| -0.0000772| -0.05499| -0.12410| 0.61232| 1.11277|

21| 0.1444| 0.781| 0.0305773| 0.0013835| -0.05245| -0.12903| 0.61223| 1.11566|

22| 0.1444| 0.988| 0.0301602| 0.0000604| -0.05093| -0.13390| 0.61359| 1.11723|

23| 0.1444| 0.798| 0.0240262| 0.0009824| -0.04901| -0.13778| 0.61364| 1.11940|

24| 0.1444| 0.968| 0.0232312| 0.0001242| -0.04778| -0.14155| 0.61461| 1.12068|

25| 0.1444| 0.813| 0.0188530| 0.0007043| -0.04632| -0.14460| 0.61472| 1.12232|

26| 0.1444| 0.952| 0.0179344| 0.0001462| -0.04532| -0.14751| 0.61542| 1.12337|

27| 0.1444| 0.825| 0.0147773| 0.0005096| -0.04421| -0.14991| 0.61555| 1.12461|

28| 0.1444| 0.939| 0.0138707| 0.0001456| -0.04342| -0.15217| 0.61605| 1.12545|

29| 0.1444| 0.835| 0.0115721| 0.0003721| -0.04256| -0.15405| 0.61619| 1.12640|

30| 0.1444| 0.929| 0.0107438| 0.0001337| -0.04193| -0.15580| 0.61656| 1.12707|

31| 0.1444| 0.843| 0.0090554| 0.0002739| -0.04128| -0.15727| 0.61669| 1.12780|

32| 0.1444| 0.921| 0.0083320| 0.0001172| -0.04078| -0.15863| 0.61696| 1.12833|

33| 0.1444| 0.850| 0.0070816| 0.0002032| -0.04027| -0.15979| 0.61707| 1.12889|

34| 0.1444| 0.914| 0.0064680| 0.0000997| -0.03988| -0.16084| 0.61727| 1.12931|

35| 0.1444| 0.856| 0.0055352| 0.0001518| -0.03949| -0.16175| 0.61736| 1.12974|

36| 0.1444| 0.908| 0.0050251| 0.0000830| -0.03918| -0.16257| 0.61751| 1.13008|

37| 0.1444| 0.861| 0.0043246| 0.0001141| -0.03888| -0.16327| 0.61759| 1.13041|

38| 0.1444| 0.904| 0.0039066| 0.0000681| -0.03863| -0.16391| 0.61770| 1.13067|

39| 0.1444| 0.865| 0.0033776| 0.0000863| -0.03840| -0.16446| 0.61777| 1.13093|

40| 0.1444| 0.900| 0.0030387| 0.0000553| -0.03821| -0.16496| 0.61786| 1.13114|

41| 0.1444| 0.868| 0.0026372| 0.0000655| -0.03803| -0.16539| 0.61791| 1.13134|

42| 0.1444| 0.897| 0.0023647| 0.0000445| -0.03788| -0.16578| 0.61797| 1.13150|

43| 0.1444| 0.871| 0.0020586| 0.0000500| -0.03774| -0.16611| 0.61802| 1.13165|

44| 0.1444| 0.894| 0.0018408| 0.0000356| -0.03762| -0.16641| 0.61807| 1.13178|

45| 0.1444| 0.873| 0.0016066| 0.0000383| -0.03751| -0.16668| 0.61810| 1.13190|

46| 0.1444| 0.892| 0.0014334| 0.0000283| -0.03742| -0.16691| 0.61814| 1.13200|

47| 0.1444| 0.875| 0.0012536| 0.0000294| -0.03733| -0.16712| 0.61816| 1.13209|

48| 0.1444| 0.891| 0.0011164| 0.0000224| -0.03726| -0.16730| 0.61819| 1.13217|

49| 0.1444| 0.876| 0.0009781| 0.0000226| -0.03719| -0.16746| 0.61822| 1.13224|

50| 0.1444| 0.889| 0.0008697| 0.0000177| -0.03714| -0.16760| 0.61824| 1.13230|

51| 0.1444| 0.877| 0.0007630| 0.0000174| -0.03709| -0.16773| 0.61825| 1.13236|

52| 0.1444| 0.888| 0.0006776| 0.0000140| -0.03704| -0.16784| 0.61827| 1.13241|

53| 0.1444| 0.878| 0.0005952| 0.0000135| -0.03700| -0.16793| 0.61829| 1.13245|

54| 0.1444| 0.887| 0.0005280| 0.0000110| -0.03697| -0.16802| 0.61830| 1.13249|

55| 0.1444| 0.879| 0.0004642| 0.0000104| -0.03694| -0.16810| 0.61831| 1.13252|

56| 0.1444| 0.886| 0.0004115| 0.0000086| -0.03691| -0.16816| 0.61832| 1.13255|

57| 0.1444| 0.880| 0.0003621| 0.0000081| -0.03689| -0.16822| 0.61833| 1.13258|

58| 0.1444| 0.886| 0.0003207| 0.0000068| -0.03687| -0.16827| 0.61834| 1.13260|

59| 0.1444| 0.880| 0.0002824| 0.0000063| -0.03685| -0.16832| 0.61834| 1.13262|

60| 0.1444| 0.885| 0.0002500| 0.0000053| -0.03683| -0.16836| 0.61835| 1.13264|

61| 0.1444| 0.881| 0.0002202| 0.0000049| -0.03682| -0.16840| 0.61835| 1.13265|

62| 0.1444| 0.885| 0.0001948| 0.0000041| -0.03680| -0.16843| 0.61836| 1.13267|

63| 0.1444| 0.881| 0.0001717| 0.0000038| -0.03679| -0.16846| 0.61836| 1.13268|

64| 0.1444| 0.885| 0.0001519| 0.0000032| -0.03678| -0.16848| 0.61837| 1.13269|

65| 0.1444| 0.882| 0.0001339| 0.0000029| -0.03677| -0.16850| 0.61837| 1.13270|

66| 0.1444| 0.884| 0.0001184| 0.0000025| -0.03676| -0.16852| 0.61837| 1.13271|

67| 0.1444| 0.882| 0.0001044| 0.0000023| -0.03676| -0.16854| 0.61837| 1.13272|

68| 0.1444| 0.884| 0.0000923| 0.0000020| -0.03675| -0.16856| 0.61838| 1.13272|

Метод наискорейшего спуска

| Норма | Оценка |

Itr| Tau | q | невязки |погрешности| x[1] | x[2] | x[3] | x[4] |

1| 0.1656| 0.662| 6.0400000| 0.3377483| 0.16556| 0.33113| 0.49669| 0.66225|

2| 0.1878| 0.439| 2.0130011| 0.3712302| -0.12546| 0.17193| 0.55639| 0.76983|

3| 0.1105| 0.440| 1.2892252| 0.1630719| -0.09434| 0.18581| 0.51451| 0.89778|

4| 0.1749| 0.591| 0.4607735| 0.0522836| -0.13659| 0.12772| 0.59017| 0.93912|

5| 0.1198| 0.720| 0.4577713| 0.0211601| -0.11963| 0.11251| 0.55818| 0.99363|

6| 0.1884| 0.954| 0.2742679| 0.0024965| -0.13156| 0.06050| 0.59840| 1.00642|

7| 0.1234| 0.629| 0.2552289| 0.0192947| -0.11210| 0.04692| 0.57620| 1.03914|

8| 0.1918| 1.243| 0.2033599| -0.0079391| -0.11352| 0.00626| 0.60441| 1.04225|

9| 0.1239| 0.557| 0.1715803| 0.0180234| -0.09701| -0.00450| 0.58724| 1.06488|

10| 0.1922| 1.376| 0.1520982| -0.0085148| -0.09624| -0.03564| 0.60810| 1.06534|

11| 0.1239| 0.535| 0.1243234| 0.0144766| -0.08325| -0.04388| 0.59494| 1.08201|

12| 0.1923| 1.423| 0.1140029| -0.0070434| -0.08222| -0.06760| 0.61066| 1.08188|

13| 0.1239| 0.529| 0.0924253| 0.0111775| -0.07223| -0.07387| 0.60063| 1.09441|

14| 0.1923| 1.438| 0.0856738| -0.0054885| -0.07135| -0.09189| 0.61254| 1.09420|

15| 0.1239| 0.527| 0.0693852| 0.0085334| -0.06373| -0.09666| 0.60492| 1.10369|

16| 0.1923| 1.443| 0.0645374| -0.0042023| -0.06303| -0.11036| 0.61395| 1.10349|

17| 0.1239| 0.526| 0.0523126| 0.0064919| -0.05725| -0.11398| 0.60816| 1.11069|

18| 0.1923| 1.444| 0.0487098| -0.0032000| -0.05671| -0.12438| 0.61502| 1.11054|

19| 0.1239| 0.526| 0.0395300| 0.0049333| -0.05231| -0.12713| 0.61062| 1.11601|

20| 0.1923| 1.445| 0.0368195| -0.0024324| -0.05190| -0.13503| 0.61583| 1.11589|

21| 0.1239| 0.526| 0.0299123| 0.0037475| -0.04856| -0.13712| 0.61249| 1.12004|

22| 0.1923| 1.445| 0.0278638| -0.0018480| -0.04824| -0.14312| 0.61645| 1.11995|

23| 0.1239| 0.526| 0.0226562| 0.0028465| -0.04571| -0.14471| 0.61391| 1.12310|

24| 0.1923| 1.445| 0.0211049| -0.0014037| -0.04547| -0.14926| 0.61692| 1.12303|

25| 0.1239| 0.526| 0.0171720| 0.0021620| -0.04354| -0.15047| 0.61499| 1.12543|

26| 0.1923| 1.445| 0.0159962| -0.0010661| -0.04336| -0.15393| 0.61727| 1.12538|

27| 0.1239| 0.526| 0.0130219| 0.0016421| -0.04190| -0.15485| 0.61581| 1.12719|

28| 0.1923| 1.445| 0.0121302| -0.0008098| -0.04176| -0.15748| 0.61754| 1.12716|

29| 0.1239| 0.526| 0.0098786| 0.0012472| -0.04065| -0.15817| 0.61643| 1.12854|

30| 0.1923| 1.445| 0.0092020| -0.0006150| -0.04055| -0.16017| 0.61775| 1.12851|

31| 0.1239| 0.526| 0.0074962| 0.0009473| -0.03970| -0.16070| 0.61690| 1.12956|

32| 0.1923| 1.445| 0.0069827| -0.0004671| -0.03962| -0.16221| 0.61790| 1.12953|

33| 0.1239| 0.526| 0.0056895| 0.0007195| -0.03898| -0.16261| 0.61726| 1.13033|

34| 0.1923| 1.445| 0.0052998| -0.0003548| -0.03892| -0.16377| 0.61802| 1.13031|

35| 0.1239| 0.526| 0.0043191| 0.0005464| -0.03843| -0.16407| 0.61753| 1.13092|

36| 0.1923| 1.445| 0.0040232| -0.0002695| -0.03839| -0.16494| 0.61811| 1.13090|

37| 0.1239| 0.526| 0.0032791| 0.0004150| -0.03802| -0.16518| 0.61774| 1.13136|

38| 0.1923| 1.445| 0.0030545| -0.0002047| -0.03798| -0.16584| 0.61818| 1.13135|

39| 0.1239| 0.526| 0.0024898| 0.0003152| -0.03770| -0.16602| 0.61790| 1.13170|

40| 0.1923| 1.445| 0.0023192| -0.0001554| -0.03768| -0.16652| 0.61823| 1.13170|

41| 0.1239| 0.526| 0.0018906| 0.0002394| -0.03746| -0.16665| 0.61802| 1.13196|

42| 0.1923| 1.445| 0.0017611| -0.0001181| -0.03744| -0.16704| 0.61827| 1.13195|

43| 0.1239| 0.526| 0.0014357| 0.0001818| -0.03728| -0.16714| 0.61811| 1.13216|

44| 0.1923| 1.445| 0.0013373| -0.0000897| -0.03727| -0.16743| 0.61830| 1.13215|

45| 0.1239| 0.526| 0.0010903| 0.0001381| -0.03714| -0.16751| 0.61818| 1.13230|

46| 0.1923| 1.445| 0.0010156| -0.0000681| -0.03713| -0.16773| 0.61832| 1.13230|

47| 0.1239| 0.526| 0.0008280| 0.0001049| -0.03704| -0.16779| 0.61823| 1.13242|

48| 0.1923| 1.445| 0.0007713| -0.0000517| -0.03703| -0.16795| 0.61834| 1.13242|

49| 0.1239| 0.526| 0.0006288| 0.0000797| -0.03696| -0.16800| 0.61827| 1.13250|

50| 0.1923| 1.445| 0.0005858| -0.0000393| -0.03695| -0.16813| 0.61835| 1.13250|

51| 0.1239| 0.526| 0.0004776| 0.0000605| -0.03690| -0.16816| 0.61830| 1.13257|

52| 0.1923| 1.445| 0.0004449| -0.0000298| -0.03689| -0.16826| 0.61836| 1.13257|

53| 0.1239| 0.526| 0.0003627| 0.0000460| -0.03685| -0.16828| 0.61832| 1.13262|

54| 0.1923| 1.445| 0.0003379| -0.0000227| -0.03685| -0.16836| 0.61837| 1.13262|

55| 0.1239| 0.526| 0.0002755| 0.0000349| -0.03682| -0.16838| 0.61834| 1.13266|

56| 0.1923| 1.445| 0.0002566| -0.0000172| -0.03681| -0.16843| 0.61837| 1.13265|

57| 0.1239| 0.526| 0.0002092| 0.0000265| -0.03679| -0.16845| 0.61835| 1.13268|

58| 0.1923| 1.445| 0.0001949| -0.0000131| -0.03679| -0.16849| 0.61838| 1.13268|

59| 0.1239| 0.526| 0.0001589| 0.0000201| -0.03677| -0.16850| 0.61836| 1.13271|

60| 0.1923| 1.445| 0.0001480| -0.0000099| -0.03677| -0.16853| 0.61838| 1.13271|

61| 0.1239| 0.526| 0.0001207| 0.0000153| -0.03675| -0.16854| 0.61837| 1.13272|

62| 0.1923| 1.445| 0.0001124| -0.0000075| -0.03675| -0.16857| 0.61839| 1.13272|

63| 0.1239| 0.526| 0.0000917| 0.0000116| -0.03674| -0.16857| 0.61837| 1.13273|

Метод ПВР - выбор оптимального w

w = 0.1 Itr = 173

w = 0.2 Itr = 84

w = 0.3 Itr = 54

w = 0.4 Itr = 40

w = 0.5 Itr = 31

w = 0.6 Itr = 25

w = 0.7 Itr = 21

w = 0.8 Itr = 18

w = 0.9 Itr = 16

w = 1.0 Itr = 13

w = 1.1 Itr = 11

w = 1.2 Itr = 9

w = 1.3 Itr = 11

w = 1.4 Itr = 11

w = 1.5 Itr = 13

w = 1.6 Itr = 18

w = 1.7 Itr = 23

w = 1.8 Itr = 34

w = 1.9 Itr = 68

w\* = 1.2 ItrMin = 9

Метод ПВР

| Норма | Оценка |

Itr| Tau | q | невязки |погрешности| x[1] | x[2] | x[3] | x[4] |

1| 1.2000| 0.712| 5.6189332| 0.2881211| 0.22222| 0.58851| 0.42882| 0.71188|

2| 1.2000| 0.846| 2.8299372| 0.1095520| -0.38010| 0.45338| 0.49201| 0.95778|

3| 1.2000| 0.540| 0.5405013| 0.2771581| -0.27947| 0.12821| 0.56254| 1.08295|

4| 1.2000| 0.586| 0.5118675| 0.1345269| -0.15132| -0.06243| 0.60084| 1.12659|

5| 1.2000| 0.423| 0.2944611| 0.1099581| -0.07696| -0.14312| 0.61512| 1.13652|

6| 1.2000| 0.383| 0.1224290| 0.0497668| -0.04604| -0.16815| 0.61898| 1.13638|

7| 1.2000| 0.303| 0.0371791| 0.0215422| -0.03666| -0.17245| 0.61933| 1.13467|

8| 1.2000| 0.160| 0.0059565| 0.0078743| -0.03516| -0.17138| 0.61895| 1.13351|

9| 1.2000| 0.961| 0.0023259| 0.0000587| -0.03564| -0.16994| 0.61863| 1.13297|

10| 1.2000| 0.573| 0.0022857| 0.0006150| -0.03622| -0.16911| 0.61847| 1.13279|

11| 1.2000| 0.411| 0.0012783| 0.0004866| -0.03654| -0.16877| 0.61841| 1.13276|

12| 1.2000| 0.382| 0.0005167| 0.0002101| -0.03667| -0.16867| 0.61839| 1.13276|

13| 1.2000| 0.293| 0.0001515| 0.0000919| -0.03671| -0.16865| 0.61839| 1.13277|

14| 1.2000| 0.148| 0.0000244| 0.0000325| -0.03671| -0.16866| 0.61839| 1.13277|

Метод сопряженных градиентов

| Норма | Оценка |

Itr| Tau | q | невязки |погрешности| x[1] | x[2] | x[3] | x[4] |

1| 0.1656| 0.662| 6.0400000| 0.3377483| 0.16556| 0.33113| 0.49669| 0.66225| 1.0000000

2| 0.1878| 0.469| 1.7428658| 0.3517847| -0.14490| 0.19858| 0.64262| 0.88915| 1.1549949

3| 0.0942| 0.411| 0.8876128| 0.1828097| -0.18897| 0.15281| 0.56999| 1.01681| 1.2040964

4| 0.6156| 2.518| 0.2347006| -0.1938245| -0.03671| -0.16867| 0.61839| 1.13277| 1.7534190

5| 0.0958| 0.000| 0.0000000| 0.3214817| -0.03671| -0.16867| 0.61839| 1.13277| 1.0000000

Число обусловленности 15.3832616

Теоретическая оценка чиcла итераций

Метод простых итераций 70

Метод наискорейшего спуска 141

Метод ПВР 9

Метод сопряженных градиентов 19

Сравнение с LU разложением

Разница между LU разложением и методом простых итераций

4.5967636009e-05 -1.1389830225e-04 1.6656322501e-05 5.0093409456e-05

Разница между LU разложением и методом наискорейшего спуска

3.6731589306e-05 -9.7785423709e-05 1.8290283373e-05 3.9468823304e-05

Разница между LU разложением и методом ПВР

7.0888678851e-06 -1.1849375698e-05 2.4064018630e-06 3.0577570191e-06

Разница между LU разложением и методом сопряженных градиентов

-1.3253287356e-15 2.5257573810e-15 -4.4408920985e-16 -6.6613381478e-16

Вариант 21b:

A:

2.5000000 -3.7000000 3.7000000 -1.2000000

-3.7000000 9.9000000 -5.5000000 3.4000000

3.7000000 -5.5000000 6.2000000 -1.3000000

-1.2000000 3.4000000 -1.3000000 7.1000000

b:

1.0000000 2.0000000 3.0000000 4.0000000

Норма матрицы 17.215255

Метод простой итерации

| Норма | Оценка |

Itr| Tau | q | невязки |погрешности| x[1] | x[2] | x[3] | x[4] |

1| 0.1046| 0.418| 9.5640307| 0.5817663| 0.10456| 0.20912| 0.31368| 0.41823|

2| 0.1046| 0.591| 4.2132183| 0.1712465| 0.19381| 0.27393| 0.56066| 0.50740|

3| 0.1046| 0.577| 2.0587722| 0.1043982| 0.20044| 0.41651| 0.66240| 0.55210|

4| 0.1046| 0.851| 1.4802488| 0.0212951| 0.22675| 0.45668| 0.78369| 0.52759|

5| 0.1046| 0.719| 0.9980536| 0.0340590| 0.21173| 0.54392| 0.83594| 0.52678|

6| 0.1046| 0.851| 0.7801840| 0.0129861| 0.21406| 0.56538| 0.91019| 0.50078|

7| 0.1046| 0.702| 0.5276397| 0.0221574| 0.19210| 0.61747| 0.94421| 0.49684|

8| 0.1046| 0.958| 0.4799096| 0.0022091| 0.18238| 0.62811| 0.99409| 0.47917|

9| 0.1046| 0.618| 0.2893314| 0.0190519| 0.15780| 0.65894| 1.01912| 0.47640|

10| 0.1046| 1.157| 0.3233767| -0.0048336| 0.14154| 0.66372| 1.05478| 0.46504|

11| 0.1046| 0.712| 0.2258268| 0.0102852| 0.11616| 0.68181| 1.07482| 0.46322|

12| 0.1046| 1.065| 0.2345596| -0.0016442| 0.09643| 0.68353| 1.10185| 0.45586|

13| 0.1046| 0.936| 0.2161092| 0.0017387| 0.07115| 0.69399| 1.11897| 0.45456|

14| 0.1046| 0.856| 0.1814900| 0.0036396| 0.04973| 0.69416| 1.14062| 0.44965|

15| 0.1046| 1.143| 0.2047204| -0.0030985| 0.02499| 0.70006| 1.15594| 0.44859|

16| 0.1046| 0.891| 0.1796602| 0.0026868| 0.00293| 0.69947| 1.17415| 0.44519|

17| 0.1046| 1.088| 0.1930743| -0.0019323| -0.02106| 0.70264| 1.18829| 0.44424|

18| 0.1046| 0.921| 0.1753872| 0.0019052| -0.04314| 0.70172| 1.20424| 0.44178|

19| 0.1046| 1.048| 0.1818028| -0.0010604| -0.06628| 0.70325| 1.21753| 0.44087|

20| 0.1046| 0.940| 0.1689062| 0.0013877| -0.08804| 0.70221| 1.23191| 0.43899|

21| 0.1046| 1.024| 0.1711653| -0.0005169| -0.11031| 0.70277| 1.24453| 0.43810|

22| 0.1046| 0.953| 0.1614212| 0.0010442| -0.13154| 0.70170| 1.25779| 0.43659|

23| 0.1046| 1.008| 0.1612419| -0.0001797| -0.15295| 0.70169| 1.26985| 0.43572|

24| 0.1046| 0.962| 0.1536049| 0.0008149| -0.17354| 0.70065| 1.28225| 0.43446|

25| 0.1046| 0.999| 0.1520289| 0.0000275| -0.19411| 0.70030| 1.29381| 0.43360|

26| 0.1046| 0.968| 0.1458276| 0.0006604| -0.21402| 0.69931| 1.30551| 0.43250|

27| 0.1046| 0.992| 0.1434860| 0.0001532| -0.23377| 0.69877| 1.31661| 0.43166|

28| 0.1046| 0.972| 0.1382864| 0.0005550| -0.25297| 0.69782| 1.32773| 0.43066|

29| 0.1046| 0.988| 0.1355602| 0.0002274| -0.27194| 0.69718| 1.33839| 0.42985|

30| 0.1046| 0.975| 0.1310792| 0.0004817| -0.29042| 0.69628| 1.34899| 0.42893|

31| 0.1046| 0.985| 0.1281965| 0.0002693| -0.30864| 0.69558| 1.35924| 0.42814|

32| 0.1046| 0.976| 0.1242476| 0.0004294| -0.32643| 0.69473| 1.36938| 0.42729|

33| 0.1046| 0.984| 0.1213431| 0.0002910| -0.34393| 0.69402| 1.37923| 0.42652|

34| 0.1046| 0.978| 0.1178018| 0.0003910| -0.36104| 0.69321| 1.38894| 0.42572|

35| 0.1046| 0.982| 0.1149529| 0.0003000| -0.37785| 0.69249| 1.39840| 0.42497|

36| 0.1046| 0.978| 0.1117345| 0.0003617| -0.39429| 0.69172| 1.40772| 0.42421|

37| 0.1046| 0.982| 0.1089840| 0.0003012| -0.41044| 0.69101| 1.41681| 0.42349|

38| 0.1046| 0.979| 0.1060298| 0.0003385| -0.42625| 0.69027| 1.42575| 0.42277|

39| 0.1046| 0.981| 0.1033993| 0.0002977| -0.44175| 0.68958| 1.43449| 0.42208|

40| 0.1046| 0.979| 0.1006673| 0.0003194| -0.45694| 0.68888| 1.44307| 0.42139|

41| 0.1046| 0.981| 0.0981661| 0.0002912| -0.47184| 0.68821| 1.45146| 0.42072|

42| 0.1046| 0.980| 0.0956256| 0.0003032| -0.48644| 0.68753| 1.45970| 0.42006|

43| 0.1046| 0.981| 0.0932551| 0.0002831| -0.50075| 0.68688| 1.46777| 0.41942|

44| 0.1046| 0.980| 0.0908830| 0.0002888| -0.51477| 0.68624| 1.47568| 0.41878|

45| 0.1046| 0.980| 0.0886407| 0.0002741| -0.52852| 0.68561| 1.48343| 0.41817|

46| 0.1046| 0.980| 0.0864188| 0.0002759| -0.54199| 0.68499| 1.49104| 0.41756|

47| 0.1046| 0.980| 0.0842997| 0.0002647| -0.55520| 0.68439| 1.49848| 0.41697|

48| 0.1046| 0.980| 0.0822134| 0.0002641| -0.56814| 0.68379| 1.50578| 0.41638|

49| 0.1046| 0.980| 0.0802115| 0.0002552| -0.58083| 0.68321| 1.51294| 0.41581|

50| 0.1046| 0.980| 0.0782488| 0.0002531| -0.59327| 0.68264| 1.51995| 0.41525|

51| 0.1046| 0.980| 0.0763574| 0.0002457| -0.60546| 0.68208| 1.52682| 0.41470|

52| 0.1046| 0.980| 0.0745081| 0.0002427| -0.61741| 0.68153| 1.53356| 0.41417|

53| 0.1046| 0.980| 0.0727206| 0.0002364| -0.62912| 0.68100| 1.54017| 0.41364|

54| 0.1046| 0.980| 0.0709758| 0.0002329| -0.64059| 0.68047| 1.54664| 0.41312|

55| 0.1046| 0.980| 0.0692858| 0.0002273| -0.65184| 0.67995| 1.55298| 0.41262|

56| 0.1046| 0.980| 0.0676379| 0.0002236| -0.66287| 0.67945| 1.55920| 0.41212|

57| 0.1046| 0.980| 0.0660391| 0.0002186| -0.67368| 0.67895| 1.56529| 0.41163|

58| 0.1046| 0.980| 0.0644811| 0.0002147| -0.68427| 0.67846| 1.57127| 0.41116|

59| 0.1046| 0.980| 0.0629679| 0.0002101| -0.69466| 0.67799| 1.57712| 0.41069|

60| 0.1046| 0.980| 0.0614936| 0.0002062| -0.70483| 0.67752| 1.58286| 0.41023|

61| 0.1046| 0.980| 0.0600605| 0.0002019| -0.71481| 0.67706| 1.58849| 0.40978|

62| 0.1046| 0.980| 0.0586644| 0.0001981| -0.72459| 0.67661| 1.59400| 0.40934|

63| 0.1046| 0.980| 0.0573063| 0.0001940| -0.73417| 0.67617| 1.59941| 0.40891|

64| 0.1046| 0.980| 0.0559833| 0.0001902| -0.74356| 0.67574| 1.60470| 0.40849|

65| 0.1046| 0.980| 0.0546956| 0.0001864| -0.75277| 0.67532| 1.60989| 0.40807|

66| 0.1046| 0.980| 0.0534409| 0.0001828| -0.76179| 0.67491| 1.61498| 0.40767|

67| 0.1046| 0.980| 0.0522193| 0.0001791| -0.77064| 0.67450| 1.61997| 0.40727|

68| 0.1046| 0.980| 0.0510288| 0.0001756| -0.77931| 0.67410| 1.62486| 0.40688|

69| 0.1046| 0.980| 0.0498691| 0.0001720| -0.78781| 0.67371| 1.62965| 0.40650|

70| 0.1046| 0.980| 0.0487389| 0.0001687| -0.79613| 0.67333| 1.63435| 0.40612|

71| 0.1046| 0.980| 0.0476375| 0.0001653| -0.80430| 0.67296| 1.63895| 0.40576|

72| 0.1046| 0.980| 0.0465639| 0.0001620| -0.81230| 0.67259| 1.64346| 0.40540|

73| 0.1046| 0.980| 0.0455173| 0.0001588| -0.82014| 0.67223| 1.64788| 0.40504|

74| 0.1046| 0.980| 0.0444969| 0.0001557| -0.82783| 0.67188| 1.65222| 0.40470|

75| 0.1046| 0.980| 0.0435020| 0.0001525| -0.83536| 0.67153| 1.65647| 0.40436|

76| 0.1046| 0.980| 0.0425317| 0.0001495| -0.84275| 0.67119| 1.66063| 0.40403|

77| 0.1046| 0.980| 0.0415855| 0.0001466| -0.84999| 0.67086| 1.66471| 0.40370|

78| 0.1046| 0.980| 0.0406624| 0.0001437| -0.85708| 0.67053| 1.66871| 0.40338|

79| 0.1046| 0.980| 0.0397620| 0.0001408| -0.86404| 0.67021| 1.67264| 0.40307|

80| 0.1046| 0.980| 0.0388836| 0.0001380| -0.87085| 0.66990| 1.67648| 0.40276|

81| 0.1046| 0.980| 0.0380264| 0.0001353| -0.87753| 0.66960| 1.68025| 0.40246|

82| 0.1046| 0.980| 0.0371900| 0.0001326| -0.88408| 0.66929| 1.68394| 0.40217|

83| 0.1046| 0.980| 0.0363738| 0.0001300| -0.89050| 0.66900| 1.68756| 0.40188|

84| 0.1046| 0.980| 0.0355771| 0.0001274| -0.89679| 0.66871| 1.69111| 0.40160|

85| 0.1046| 0.980| 0.0347995| 0.0001248| -0.90295| 0.66843| 1.69458| 0.40132|

86| 0.1046| 0.980| 0.0340405| 0.0001224| -0.90900| 0.66815| 1.69799| 0.40105|

87| 0.1046| 0.980| 0.0332994| 0.0001199| -0.91492| 0.66788| 1.70133| 0.40078|

88| 0.1046| 0.980| 0.0325758| 0.0001176| -0.92073| 0.66761| 1.70460| 0.40052|

89| 0.1046| 0.980| 0.0318694| 0.0001152| -0.92642| 0.66735| 1.70781| 0.40026|

90| 0.1046| 0.980| 0.0311795| 0.0001129| -0.93200| 0.66710| 1.71096| 0.40001|

91| 0.1046| 0.980| 0.0305057| 0.0001107| -0.93746| 0.66684| 1.71404| 0.39976|

92| 0.1046| 0.980| 0.0298477| 0.0001085| -0.94282| 0.66660| 1.71706| 0.39952|

93| 0.1046| 0.980| 0.0292050| 0.0001063| -0.94807| 0.66636| 1.72002| 0.39929|

94| 0.1046| 0.980| 0.0285771| 0.0001042| -0.95322| 0.66612| 1.72293| 0.39906|

95| 0.1046| 0.980| 0.0279638| 0.0001022| -0.95827| 0.66589| 1.72577| 0.39883|

96| 0.1046| 0.980| 0.0273646| 0.0001001| -0.96321| 0.66566| 1.72856| 0.39861|

97| 0.1046| 0.980| 0.0267792| 0.0000982| -0.96806| 0.66544| 1.73130| 0.39839|

98| 0.1046| 0.980| 0.0262072| 0.0000962| -0.97281| 0.66522| 1.73397| 0.39817|

99| 0.1046| 0.980| 0.0256482| 0.0000943| -0.97747| 0.66501| 1.73660| 0.39796|

100| 0.1046| 0.980| 0.0251020| 0.0000924| -0.98204| 0.66480| 1.73917| 0.39776|

101| 0.1046| 0.980| 0.0245682| 0.0000906| -0.98651| 0.66459| 1.74170| 0.39756|

102| 0.1046| 0.980| 0.0240465| 0.0000888| -0.99089| 0.66439| 1.74417| 0.39736|

103| 0.1046| 0.980| 0.0235366| 0.0000870| -0.99519| 0.66420| 1.74659| 0.39717|

104| 0.1046| 0.980| 0.0230381| 0.0000853| -0.99941| 0.66400| 1.74897| 0.39698|

105| 0.1046| 0.980| 0.0225509| 0.0000836| -1.00354| 0.66381| 1.75130| 0.39679|

106| 0.1046| 0.980| 0.0220746| 0.0000820| -1.00758| 0.66363| 1.75358| 0.39661|

107| 0.1046| 0.980| 0.0216089| 0.0000803| -1.01155| 0.66344| 1.75582| 0.39643|

108| 0.1046| 0.980| 0.0211537| 0.0000787| -1.01544| 0.66327| 1.75801| 0.39626|

109| 0.1046| 0.980| 0.0207085| 0.0000772| -1.01925| 0.66309| 1.76016| 0.39609|

110| 0.1046| 0.980| 0.0202733| 0.0000756| -1.02298| 0.66292| 1.76227| 0.39592|

111| 0.1046| 0.980| 0.0198477| 0.0000741| -1.02665| 0.66275| 1.76433| 0.39575|

112| 0.1046| 0.980| 0.0194315| 0.0000727| -1.03023| 0.66259| 1.76635| 0.39559|

113| 0.1046| 0.980| 0.0190245| 0.0000712| -1.03375| 0.66243| 1.76834| 0.39543|

114| 0.1046| 0.980| 0.0186265| 0.0000698| -1.03720| 0.66227| 1.77028| 0.39528|

115| 0.1046| 0.980| 0.0182372| 0.0000684| -1.04058| 0.66211| 1.77219| 0.39513|

116| 0.1046| 0.980| 0.0178564| 0.0000671| -1.04389| 0.66196| 1.77406| 0.39498|

117| 0.1046| 0.980| 0.0174840| 0.0000657| -1.04714| 0.66181| 1.77589| 0.39483|

118| 0.1046| 0.980| 0.0171197| 0.0000644| -1.05032| 0.66167| 1.77768| 0.39469|

119| 0.1046| 0.980| 0.0167634| 0.0000632| -1.05344| 0.66152| 1.77944| 0.39455|

120| 0.1046| 0.980| 0.0164148| 0.0000619| -1.05650| 0.66138| 1.78116| 0.39441|

121| 0.1046| 0.980| 0.0160738| 0.0000607| -1.05949| 0.66124| 1.78285| 0.39427|

122| 0.1046| 0.980| 0.0157402| 0.0000595| -1.06243| 0.66111| 1.78451| 0.39414|

123| 0.1046| 0.980| 0.0154138| 0.0000583| -1.06531| 0.66098| 1.78613| 0.39401|

124| 0.1046| 0.980| 0.0150944| 0.0000571| -1.06813| 0.66085| 1.78772| 0.39389|

125| 0.1046| 0.980| 0.0147820| 0.0000560| -1.07089| 0.66072| 1.78928| 0.39376|

126| 0.1046| 0.980| 0.0144763| 0.0000549| -1.07361| 0.66060| 1.79081| 0.39364|

127| 0.1046| 0.980| 0.0141771| 0.0000538| -1.07626| 0.66048| 1.79231| 0.39352|

128| 0.1046| 0.980| 0.0138844| 0.0000527| -1.07887| 0.66036| 1.79378| 0.39340|

129| 0.1046| 0.980| 0.0135979| 0.0000517| -1.08142| 0.66024| 1.79522| 0.39329|

130| 0.1046| 0.980| 0.0133176| 0.0000507| -1.08392| 0.66012| 1.79663| 0.39318|

131| 0.1046| 0.980| 0.0130433| 0.0000496| -1.08637| 0.66001| 1.79801| 0.39307|

132| 0.1046| 0.980| 0.0127748| 0.0000487| -1.08878| 0.65990| 1.79937| 0.39296|

133| 0.1046| 0.980| 0.0125120| 0.0000477| -1.09113| 0.65979| 1.80069| 0.39285|

134| 0.1046| 0.980| 0.0122548| 0.0000468| -1.09344| 0.65969| 1.80200| 0.39275|

135| 0.1046| 0.980| 0.0120031| 0.0000458| -1.09570| 0.65958| 1.80327| 0.39265|

136| 0.1046| 0.980| 0.0117568| 0.0000449| -1.09792| 0.65948| 1.80452| 0.39255|

137| 0.1046| 0.980| 0.0115156| 0.0000440| -1.10010| 0.65938| 1.80575| 0.39245|

138| 0.1046| 0.980| 0.0112796| 0.0000431| -1.10223| 0.65928| 1.80695| 0.39235|

139| 0.1046| 0.980| 0.0110485| 0.0000423| -1.10432| 0.65919| 1.80813| 0.39226|

140| 0.1046| 0.980| 0.0108223| 0.0000415| -1.10636| 0.65909| 1.80928| 0.39217|

141| 0.1046| 0.980| 0.0106009| 0.0000406| -1.10837| 0.65900| 1.81041| 0.39208|

142| 0.1046| 0.980| 0.0103842| 0.0000398| -1.11034| 0.65891| 1.81152| 0.39199|

143| 0.1046| 0.980| 0.0101720| 0.0000390| -1.11227| 0.65882| 1.81261| 0.39190|

144| 0.1046| 0.980| 0.0099643| 0.0000383| -1.11415| 0.65874| 1.81368| 0.39182|

145| 0.1046| 0.980| 0.0097609| 0.0000375| -1.11601| 0.65865| 1.81472| 0.39173|

146| 0.1046| 0.980| 0.0095618| 0.0000368| -1.11782| 0.65857| 1.81574| 0.39165|

147| 0.1046| 0.980| 0.0093669| 0.0000360| -1.11960| 0.65849| 1.81675| 0.39157|

148| 0.1046| 0.980| 0.0091760| 0.0000353| -1.12135| 0.65841| 1.81773| 0.39149|

149| 0.1046| 0.980| 0.0089891| 0.0000346| -1.12305| 0.65833| 1.81869| 0.39141|

150| 0.1046| 0.980| 0.0088061| 0.0000339| -1.12473| 0.65825| 1.81964| 0.39134|

151| 0.1046| 0.980| 0.0086270| 0.0000333| -1.12637| 0.65818| 1.82057| 0.39127|

152| 0.1046| 0.980| 0.0084516| 0.0000326| -1.12798| 0.65810| 1.82147| 0.39119|

153| 0.1046| 0.980| 0.0082798| 0.0000319| -1.12956| 0.65803| 1.82236| 0.39112|

154| 0.1046| 0.980| 0.0081116| 0.0000313| -1.13111| 0.65796| 1.82323| 0.39105|

155| 0.1046| 0.980| 0.0079469| 0.0000307| -1.13262| 0.65789| 1.82409| 0.39098|

156| 0.1046| 0.980| 0.0077856| 0.0000301| -1.13411| 0.65782| 1.82493| 0.39092|

157| 0.1046| 0.980| 0.0076277| 0.0000295| -1.13556| 0.65775| 1.82575| 0.39085|

158| 0.1046| 0.980| 0.0074730| 0.0000289| -1.13699| 0.65769| 1.82655| 0.39079|

159| 0.1046| 0.980| 0.0073215| 0.0000283| -1.13839| 0.65762| 1.82734| 0.39072|

160| 0.1046| 0.980| 0.0071732| 0.0000278| -1.13976| 0.65756| 1.82812| 0.39066|

161| 0.1046| 0.980| 0.0070279| 0.0000272| -1.14110| 0.65750| 1.82887| 0.39060|

162| 0.1046| 0.980| 0.0068856| 0.0000267| -1.14242| 0.65744| 1.82962| 0.39054|

163| 0.1046| 0.980| 0.0067463| 0.0000261| -1.14371| 0.65738| 1.83034| 0.39049|

164| 0.1046| 0.980| 0.0066098| 0.0000256| -1.14498| 0.65732| 1.83106| 0.39043|

165| 0.1046| 0.980| 0.0064762| 0.0000251| -1.14622| 0.65726| 1.83176| 0.39037|

166| 0.1046| 0.980| 0.0063453| 0.0000246| -1.14743| 0.65721| 1.83244| 0.39032|

167| 0.1046| 0.980| 0.0062171| 0.0000241| -1.14863| 0.65715| 1.83311| 0.39026|

168| 0.1046| 0.980| 0.0060915| 0.0000236| -1.14979| 0.65710| 1.83377| 0.39021|

169| 0.1046| 0.980| 0.0059685| 0.0000232| -1.15094| 0.65705| 1.83442| 0.39016|

170| 0.1046| 0.980| 0.0058480| 0.0000227| -1.15206| 0.65700| 1.83505| 0.39011|

171| 0.1046| 0.980| 0.0057301| 0.0000223| -1.15316| 0.65695| 1.83567| 0.39006|

172| 0.1046| 0.980| 0.0056145| 0.0000218| -1.15424| 0.65690| 1.83628| 0.39001|

173| 0.1046| 0.980| 0.0055013| 0.0000214| -1.15530| 0.65685| 1.83688| 0.38996|

174| 0.1046| 0.980| 0.0053904| 0.0000210| -1.15633| 0.65680| 1.83746| 0.38992|

175| 0.1046| 0.980| 0.0052818| 0.0000206| -1.15735| 0.65675| 1.83803| 0.38987|

176| 0.1046| 0.980| 0.0051754| 0.0000201| -1.15834| 0.65671| 1.83859| 0.38983|

177| 0.1046| 0.980| 0.0050711| 0.0000197| -1.15932| 0.65666| 1.83914| 0.38978|

178| 0.1046| 0.980| 0.0049690| 0.0000194| -1.16027| 0.65662| 1.83968| 0.38974|

179| 0.1046| 0.980| 0.0048690| 0.0000190| -1.16121| 0.65658| 1.84021| 0.38970|

180| 0.1046| 0.980| 0.0047710| 0.0000186| -1.16213| 0.65653| 1.84073| 0.38966|

181| 0.1046| 0.980| 0.0046751| 0.0000182| -1.16303| 0.65649| 1.84123| 0.38962|

182| 0.1046| 0.980| 0.0045811| 0.0000179| -1.16391| 0.65645| 1.84173| 0.38958|

183| 0.1046| 0.980| 0.0044889| 0.0000175| -1.16477| 0.65641| 1.84222| 0.38954|

184| 0.1046| 0.980| 0.0043987| 0.0000172| -1.16562| 0.65637| 1.84270| 0.38950|

185| 0.1046| 0.980| 0.0043103| 0.0000168| -1.16645| 0.65634| 1.84317| 0.38946|

186| 0.1046| 0.980| 0.0042237| 0.0000165| -1.16727| 0.65630| 1.84363| 0.38943|

187| 0.1046| 0.980| 0.0041389| 0.0000162| -1.16806| 0.65626| 1.84408| 0.38939|

188| 0.1046| 0.980| 0.0040558| 0.0000158| -1.16885| 0.65623| 1.84452| 0.38935|

189| 0.1046| 0.980| 0.0039744| 0.0000155| -1.16961| 0.65619| 1.84495| 0.38932|

190| 0.1046| 0.980| 0.0038946| 0.0000152| -1.17036| 0.65616| 1.84537| 0.38929|

191| 0.1046| 0.980| 0.0038164| 0.0000149| -1.17110| 0.65612| 1.84579| 0.38925|

192| 0.1046| 0.980| 0.0037399| 0.0000146| -1.17182| 0.65609| 1.84619| 0.38922|

193| 0.1046| 0.980| 0.0036649| 0.0000143| -1.17253| 0.65606| 1.84659| 0.38919|

194| 0.1046| 0.980| 0.0035914| 0.0000140| -1.17322| 0.65603| 1.84699| 0.38916|

195| 0.1046| 0.980| 0.0035194| 0.0000138| -1.17390| 0.65599| 1.84737| 0.38913|

196| 0.1046| 0.980| 0.0034488| 0.0000135| -1.17457| 0.65596| 1.84774| 0.38910|

197| 0.1046| 0.980| 0.0033797| 0.0000132| -1.17522| 0.65593| 1.84811| 0.38907|

198| 0.1046| 0.980| 0.0033120| 0.0000130| -1.17586| 0.65590| 1.84847| 0.38904|

199| 0.1046| 0.980| 0.0032456| 0.0000127| -1.17649| 0.65588| 1.84883| 0.38901|

200| 0.1046| 0.980| 0.0031806| 0.0000125| -1.17711| 0.65585| 1.84917| 0.38898|

201| 0.1046| 0.980| 0.0031169| 0.0000122| -1.17771| 0.65582| 1.84951| 0.38896|

202| 0.1046| 0.980| 0.0030545| 0.0000120| -1.17830| 0.65579| 1.84985| 0.38893|

203| 0.1046| 0.980| 0.0029934| 0.0000117| -1.17888| 0.65577| 1.85017| 0.38890|

204| 0.1046| 0.980| 0.0029335| 0.0000115| -1.17945| 0.65574| 1.85049| 0.38888|

205| 0.1046| 0.980| 0.0028748| 0.0000113| -1.18000| 0.65571| 1.85081| 0.38885|

206| 0.1046| 0.980| 0.0028172| 0.0000110| -1.18055| 0.65569| 1.85111| 0.38883|

207| 0.1046| 0.980| 0.0027609| 0.0000108| -1.18108| 0.65566| 1.85142| 0.38880|

208| 0.1046| 0.980| 0.0027057| 0.0000106| -1.18161| 0.65564| 1.85171| 0.38878|

209| 0.1046| 0.980| 0.0026515| 0.0000104| -1.18212| 0.65562| 1.85200| 0.38876|

210| 0.1046| 0.980| 0.0025985| 0.0000102| -1.18262| 0.65559| 1.85228| 0.38873|

211| 0.1046| 0.980| 0.0025466| 0.0000100| -1.18312| 0.65557| 1.85256| 0.38871|

212| 0.1046| 0.980| 0.0024957| 0.0000098| -1.18360| 0.65555| 1.85284| 0.38869|

213| 0.1046| 0.980| 0.0024458| 0.0000096| -1.18407| 0.65553| 1.85310| 0.38867|

214| 0.1046| 0.980| 0.0023969| 0.0000094| -1.18454| 0.65551| 1.85336| 0.38865|

215| 0.1046| 0.980| 0.0023490| 0.0000092| -1.18499| 0.65549| 1.85362| 0.38863|

216| 0.1046| 0.980| 0.0023021| 0.0000090| -1.18544| 0.65546| 1.85387| 0.38861|

217| 0.1046| 0.980| 0.0022561| 0.0000089| -1.18588| 0.65544| 1.85412| 0.38859|

218| 0.1046| 0.980| 0.0022110| 0.0000087| -1.18631| 0.65543| 1.85436| 0.38857|

219| 0.1046| 0.980| 0.0021669| 0.0000085| -1.18673| 0.65541| 1.85460| 0.38855|

220| 0.1046| 0.980| 0.0021236| 0.0000083| -1.18714| 0.65539| 1.85483| 0.38853|

221| 0.1046| 0.980| 0.0020812| 0.0000082| -1.18754| 0.65537| 1.85506| 0.38851|

222| 0.1046| 0.980| 0.0020397| 0.0000080| -1.18794| 0.65535| 1.85528| 0.38850|

223| 0.1046| 0.980| 0.0019989| 0.0000079| -1.18832| 0.65533| 1.85550| 0.38848|

224| 0.1046| 0.980| 0.0019590| 0.0000077| -1.18870| 0.65532| 1.85571| 0.38846|

225| 0.1046| 0.980| 0.0019200| 0.0000075| -1.18908| 0.65530| 1.85592| 0.38844|

226| 0.1046| 0.980| 0.0018816| 0.0000074| -1.18944| 0.65528| 1.85613| 0.38843|

227| 0.1046| 0.980| 0.0018441| 0.0000072| -1.18980| 0.65526| 1.85633| 0.38841|

228| 0.1046| 0.980| 0.0018073| 0.0000071| -1.19015| 0.65525| 1.85653| 0.38840|

229| 0.1046| 0.980| 0.0017713| 0.0000070| -1.19049| 0.65523| 1.85672| 0.38838|

230| 0.1046| 0.980| 0.0017359| 0.0000068| -1.19083| 0.65522| 1.85691| 0.38837|

231| 0.1046| 0.980| 0.0017013| 0.0000067| -1.19116| 0.65520| 1.85710| 0.38835|

232| 0.1046| 0.980| 0.0016674| 0.0000066| -1.19149| 0.65519| 1.85728| 0.38834|

233| 0.1046| 0.980| 0.0016341| 0.0000064| -1.19180| 0.65517| 1.85746| 0.38832|

234| 0.1046| 0.980| 0.0016015| 0.0000063| -1.19211| 0.65516| 1.85764| 0.38831|

235| 0.1046| 0.980| 0.0015696| 0.0000062| -1.19242| 0.65514| 1.85781| 0.38829|

236| 0.1046| 0.980| 0.0015383| 0.0000061| -1.19272| 0.65513| 1.85798| 0.38828|

237| 0.1046| 0.980| 0.0015077| 0.0000059| -1.19301| 0.65512| 1.85814| 0.38827|

238| 0.1046| 0.980| 0.0014776| 0.0000058| -1.19330| 0.65510| 1.85830| 0.38825|

239| 0.1046| 0.980| 0.0014482| 0.0000057| -1.19358| 0.65509| 1.85846| 0.38824|

240| 0.1046| 0.980| 0.0014193| 0.0000056| -1.19386| 0.65508| 1.85862| 0.38823|

241| 0.1046| 0.980| 0.0013910| 0.0000055| -1.19413| 0.65507| 1.85877| 0.38822|

242| 0.1046| 0.980| 0.0013633| 0.0000054| -1.19439| 0.65505| 1.85892| 0.38821|

243| 0.1046| 0.980| 0.0013361| 0.0000053| -1.19465| 0.65504| 1.85907| 0.38819|

244| 0.1046| 0.980| 0.0013095| 0.0000052| -1.19491| 0.65503| 1.85921| 0.38818|

245| 0.1046| 0.980| 0.0012834| 0.0000051| -1.19515| 0.65502| 1.85935| 0.38817|

246| 0.1046| 0.980| 0.0012579| 0.0000050| -1.19540| 0.65501| 1.85949| 0.38816|

247| 0.1046| 0.980| 0.0012328| 0.0000049| -1.19564| 0.65500| 1.85962| 0.38815|

248| 0.1046| 0.980| 0.0012083| 0.0000048| -1.19587| 0.65499| 1.85976| 0.38814|

249| 0.1046| 0.980| 0.0011842| 0.0000047| -1.19610| 0.65498| 1.85989| 0.38813|

250| 0.1046| 0.980| 0.0011606| 0.0000046| -1.19633| 0.65497| 1.86001| 0.38812|

251| 0.1046| 0.980| 0.0011375| 0.0000045| -1.19655| 0.65495| 1.86014| 0.38811|

252| 0.1046| 0.980| 0.0011149| 0.0000044| -1.19677| 0.65495| 1.86026| 0.38810|

253| 0.1046| 0.980| 0.0010927| 0.0000043| -1.19698| 0.65494| 1.86038| 0.38809|

254| 0.1046| 0.980| 0.0010709| 0.0000042| -1.19719| 0.65493| 1.86050| 0.38808|

255| 0.1046| 0.980| 0.0010496| 0.0000041| -1.19739| 0.65492| 1.86061| 0.38807|

256| 0.1046| 0.980| 0.0010287| 0.0000041| -1.19759| 0.65491| 1.86073| 0.38806|

257| 0.1046| 0.980| 0.0010082| 0.0000040| -1.19779| 0.65490| 1.86084| 0.38805|

258| 0.1046| 0.980| 0.0009882| 0.0000039| -1.19798| 0.65489| 1.86095| 0.38804|

259| 0.1046| 0.980| 0.0009685| 0.0000038| -1.19817| 0.65488| 1.86105| 0.38804|

260| 0.1046| 0.980| 0.0009492| 0.0000037| -1.19835| 0.65487| 1.86116| 0.38803|

261| 0.1046| 0.980| 0.0009303| 0.0000037| -1.19854| 0.65486| 1.86126| 0.38802|

262| 0.1046| 0.980| 0.0009118| 0.0000036| -1.19871| 0.65486| 1.86136| 0.38801|

263| 0.1046| 0.980| 0.0008937| 0.0000035| -1.19889| 0.65485| 1.86146| 0.38800|

264| 0.1046| 0.980| 0.0008759| 0.0000035| -1.19906| 0.65484| 1.86155| 0.38800|

265| 0.1046| 0.980| 0.0008585| 0.0000034| -1.19922| 0.65483| 1.86165| 0.38799|

266| 0.1046| 0.980| 0.0008414| 0.0000033| -1.19939| 0.65482| 1.86174| 0.38798|

267| 0.1046| 0.980| 0.0008247| 0.0000033| -1.19955| 0.65482| 1.86183| 0.38797|

268| 0.1046| 0.980| 0.0008083| 0.0000032| -1.19971| 0.65481| 1.86192| 0.38797|

269| 0.1046| 0.980| 0.0007922| 0.0000031| -1.19986| 0.65480| 1.86201| 0.38796|

270| 0.1046| 0.980| 0.0007764| 0.0000031| -1.20001| 0.65480| 1.86209| 0.38795|

271| 0.1046| 0.980| 0.0007610| 0.0000030| -1.20016| 0.65479| 1.86217| 0.38795|

272| 0.1046| 0.980| 0.0007458| 0.0000029| -1.20031| 0.65478| 1.86226| 0.38794|

273| 0.1046| 0.980| 0.0007310| 0.0000029| -1.20045| 0.65478| 1.86234| 0.38793|

274| 0.1046| 0.980| 0.0007165| 0.0000028| -1.20059| 0.65477| 1.86241| 0.38793|

275| 0.1046| 0.980| 0.0007022| 0.0000028| -1.20072| 0.65476| 1.86249| 0.38792|

276| 0.1046| 0.980| 0.0006883| 0.0000027| -1.20086| 0.65476| 1.86257| 0.38791|

277| 0.1046| 0.980| 0.0006746| 0.0000027| -1.20099| 0.65475| 1.86264| 0.38791|

278| 0.1046| 0.980| 0.0006612| 0.0000026| -1.20112| 0.65475| 1.86271| 0.38790|

279| 0.1046| 0.980| 0.0006480| 0.0000026| -1.20124| 0.65474| 1.86279| 0.38790|

280| 0.1046| 0.980| 0.0006351| 0.0000025| -1.20137| 0.65473| 1.86286| 0.38789|

281| 0.1046| 0.980| 0.0006225| 0.0000025| -1.20149| 0.65473| 1.86292| 0.38789|

282| 0.1046| 0.980| 0.0006101| 0.0000024| -1.20161| 0.65472| 1.86299| 0.38788|

283| 0.1046| 0.980| 0.0005980| 0.0000024| -1.20172| 0.65472| 1.86306| 0.38788|

284| 0.1046| 0.980| 0.0005861| 0.0000023| -1.20184| 0.65471| 1.86312| 0.38787|

285| 0.1046| 0.980| 0.0005745| 0.0000023| -1.20195| 0.65471| 1.86318| 0.38786|

286| 0.1046| 0.980| 0.0005630| 0.0000022| -1.20206| 0.65470| 1.86325| 0.38786|

287| 0.1046| 0.980| 0.0005518| 0.0000022| -1.20217| 0.65470| 1.86331| 0.38786|

288| 0.1046| 0.980| 0.0005409| 0.0000021| -1.20227| 0.65469| 1.86337| 0.38785|

289| 0.1046| 0.980| 0.0005301| 0.0000021| -1.20238| 0.65469| 1.86342| 0.38785|

290| 0.1046| 0.980| 0.0005196| 0.0000020| -1.20248| 0.65468| 1.86348| 0.38784|

291| 0.1046| 0.980| 0.0005093| 0.0000020| -1.20258| 0.65468| 1.86354| 0.38784|

292| 0.1046| 0.980| 0.0004991| 0.0000020| -1.20267| 0.65467| 1.86359| 0.38783|

293| 0.1046| 0.980| 0.0004892| 0.0000019| -1.20277| 0.65467| 1.86365| 0.38783|

294| 0.1046| 0.980| 0.0004795| 0.0000019| -1.20286| 0.65467| 1.86370| 0.38782|

295| 0.1046| 0.980| 0.0004700| 0.0000019| -1.20295| 0.65466| 1.86375| 0.38782|

296| 0.1046| 0.980| 0.0004606| 0.0000018| -1.20304| 0.65466| 1.86380| 0.38782|

297| 0.1046| 0.980| 0.0004515| 0.0000018| -1.20313| 0.65465| 1.86385| 0.38781|

298| 0.1046| 0.980| 0.0004425| 0.0000017| -1.20322| 0.65465| 1.86390| 0.38781|

299| 0.1046| 0.980| 0.0004337| 0.0000017| -1.20330| 0.65465| 1.86395| 0.38780|

300| 0.1046| 0.980| 0.0004251| 0.0000017| -1.20339| 0.65464| 1.86399| 0.38780|

301| 0.1046| 0.980| 0.0004166| 0.0000016| -1.20347| 0.65464| 1.86404| 0.38780|

302| 0.1046| 0.980| 0.0004084| 0.0000016| -1.20355| 0.65463| 1.86408| 0.38779|

303| 0.1046| 0.980| 0.0004003| 0.0000016| -1.20362| 0.65463| 1.86413| 0.38779|

304| 0.1046| 0.980| 0.0003923| 0.0000015| -1.20370| 0.65463| 1.86417| 0.38779|

305| 0.1046| 0.980| 0.0003845| 0.0000015| -1.20378| 0.65462| 1.86421| 0.38778|

306| 0.1046| 0.980| 0.0003769| 0.0000015| -1.20385| 0.65462| 1.86425| 0.38778|

307| 0.1046| 0.980| 0.0003694| 0.0000015| -1.20392| 0.65462| 1.86430| 0.38778|

308| 0.1046| 0.980| 0.0003620| 0.0000014| -1.20399| 0.65461| 1.86434| 0.38777|

309| 0.1046| 0.980| 0.0003549| 0.0000014| -1.20406| 0.65461| 1.86437| 0.38777|

310| 0.1046| 0.980| 0.0003478| 0.0000014| -1.20413| 0.65461| 1.86441| 0.38777|

311| 0.1046| 0.980| 0.0003409| 0.0000013| -1.20420| 0.65460| 1.86445| 0.38776|

312| 0.1046| 0.980| 0.0003341| 0.0000013| -1.20426| 0.65460| 1.86449| 0.38776|

313| 0.1046| 0.980| 0.0003275| 0.0000013| -1.20432| 0.65460| 1.86452| 0.38776|

314| 0.1046| 0.980| 0.0003210| 0.0000013| -1.20439| 0.65460| 1.86456| 0.38776|

315| 0.1046| 0.980| 0.0003146| 0.0000012| -1.20445| 0.65459| 1.86459| 0.38775|

316| 0.1046| 0.980| 0.0003084| 0.0000012| -1.20451| 0.65459| 1.86463| 0.38775|

317| 0.1046| 0.980| 0.0003022| 0.0000012| -1.20457| 0.65459| 1.86466| 0.38775|

318| 0.1046| 0.980| 0.0002962| 0.0000012| -1.20463| 0.65458| 1.86469| 0.38774|

319| 0.1046| 0.980| 0.0002903| 0.0000011| -1.20468| 0.65458| 1.86472| 0.38774|

320| 0.1046| 0.980| 0.0002846| 0.0000011| -1.20474| 0.65458| 1.86476| 0.38774|

321| 0.1046| 0.980| 0.0002789| 0.0000011| -1.20479| 0.65458| 1.86479| 0.38774|

322| 0.1046| 0.980| 0.0002734| 0.0000011| -1.20485| 0.65457| 1.86482| 0.38773|

323| 0.1046| 0.980| 0.0002680| 0.0000011| -1.20490| 0.65457| 1.86485| 0.38773|

324| 0.1046| 0.980| 0.0002626| 0.0000010| -1.20495| 0.65457| 1.86487| 0.38773|

325| 0.1046| 0.980| 0.0002574| 0.0000010| -1.20500| 0.65457| 1.86490| 0.38773|

326| 0.1046| 0.980| 0.0002523| 0.0000010| -1.20505| 0.65457| 1.86493| 0.38773|

327| 0.1046| 0.980| 0.0002473| 0.0000010| -1.20510| 0.65456| 1.86496| 0.38772|

328| 0.1046| 0.980| 0.0002424| 0.0000010| -1.20514| 0.65456| 1.86498| 0.38772|

329| 0.1046| 0.980| 0.0002376| 0.0000009| -1.20519| 0.65456| 1.86501| 0.38772|

330| 0.1046| 0.980| 0.0002328| 0.0000009| -1.20524| 0.65456| 1.86504| 0.38772|

331| 0.1046| 0.980| 0.0002282| 0.0000009| -1.20528| 0.65455| 1.86506| 0.38772|

332| 0.1046| 0.980| 0.0002237| 0.0000009| -1.20532| 0.65455| 1.86509| 0.38771|

333| 0.1046| 0.980| 0.0002193| 0.0000009| -1.20537| 0.65455| 1.86511| 0.38771|

334| 0.1046| 0.980| 0.0002149| 0.0000008| -1.20541| 0.65455| 1.86513| 0.38771|

335| 0.1046| 0.980| 0.0002106| 0.0000008| -1.20545| 0.65455| 1.86516| 0.38771|

336| 0.1046| 0.980| 0.0002064| 0.0000008| -1.20549| 0.65454| 1.86518| 0.38771|

337| 0.1046| 0.980| 0.0002023| 0.0000008| -1.20553| 0.65454| 1.86520| 0.38770|

338| 0.1046| 0.980| 0.0001983| 0.0000008| -1.20557| 0.65454| 1.86522| 0.38770|

339| 0.1046| 0.980| 0.0001944| 0.0000008| -1.20561| 0.65454| 1.86524| 0.38770|

340| 0.1046| 0.980| 0.0001905| 0.0000008| -1.20564| 0.65454| 1.86527| 0.38770|

341| 0.1046| 0.980| 0.0001867| 0.0000007| -1.20568| 0.65454| 1.86529| 0.38770|

342| 0.1046| 0.980| 0.0001830| 0.0000007| -1.20571| 0.65453| 1.86531| 0.38770|

343| 0.1046| 0.980| 0.0001794| 0.0000007| -1.20575| 0.65453| 1.86533| 0.38769|

344| 0.1046| 0.980| 0.0001758| 0.0000007| -1.20578| 0.65453| 1.86535| 0.38769|

345| 0.1046| 0.980| 0.0001724| 0.0000007| -1.20582| 0.65453| 1.86536| 0.38769|

346| 0.1046| 0.980| 0.0001689| 0.0000007| -1.20585| 0.65453| 1.86538| 0.38769|

347| 0.1046| 0.980| 0.0001656| 0.0000007| -1.20588| 0.65453| 1.86540| 0.38769|

348| 0.1046| 0.980| 0.0001623| 0.0000006| -1.20591| 0.65453| 1.86542| 0.38769|

349| 0.1046| 0.980| 0.0001591| 0.0000006| -1.20595| 0.65452| 1.86544| 0.38769|

350| 0.1046| 0.980| 0.0001559| 0.0000006| -1.20598| 0.65452| 1.86545| 0.38768|

351| 0.1046| 0.980| 0.0001528| 0.0000006| -1.20601| 0.65452| 1.86547| 0.38768|

352| 0.1046| 0.980| 0.0001498| 0.0000006| -1.20603| 0.65452| 1.86549| 0.38768|

353| 0.1046| 0.980| 0.0001468| 0.0000006| -1.20606| 0.65452| 1.86550| 0.38768|

354| 0.1046| 0.980| 0.0001439| 0.0000006| -1.20609| 0.65452| 1.86552| 0.38768|

355| 0.1046| 0.980| 0.0001410| 0.0000006| -1.20612| 0.65452| 1.86553| 0.38768|

356| 0.1046| 0.980| 0.0001382| 0.0000005| -1.20615| 0.65451| 1.86555| 0.38768|

357| 0.1046| 0.980| 0.0001355| 0.0000005| -1.20617| 0.65451| 1.86556| 0.38767|

358| 0.1046| 0.980| 0.0001328| 0.0000005| -1.20620| 0.65451| 1.86558| 0.38767|

359| 0.1046| 0.980| 0.0001302| 0.0000005| -1.20622| 0.65451| 1.86559| 0.38767|

360| 0.1046| 0.980| 0.0001276| 0.0000005| -1.20625| 0.65451| 1.86561| 0.38767|

361| 0.1046| 0.980| 0.0001250| 0.0000005| -1.20627| 0.65451| 1.86562| 0.38767|

362| 0.1046| 0.980| 0.0001226| 0.0000005| -1.20630| 0.65451| 1.86563| 0.38767|

363| 0.1046| 0.980| 0.0001201| 0.0000005| -1.20632| 0.65451| 1.86565| 0.38767|

364| 0.1046| 0.980| 0.0001177| 0.0000005| -1.20634| 0.65451| 1.86566| 0.38767|

365| 0.1046| 0.980| 0.0001154| 0.0000005| -1.20637| 0.65450| 1.86567| 0.38767|

366| 0.1046| 0.980| 0.0001131| 0.0000004| -1.20639| 0.65450| 1.86569| 0.38767|

367| 0.1046| 0.980| 0.0001109| 0.0000004| -1.20641| 0.65450| 1.86570| 0.38766|

368| 0.1046| 0.980| 0.0001087| 0.0000004| -1.20643| 0.65450| 1.86571| 0.38766|

369| 0.1046| 0.980| 0.0001065| 0.0000004| -1.20645| 0.65450| 1.86572| 0.38766|

370| 0.1046| 0.980| 0.0001044| 0.0000004| -1.20647| 0.65450| 1.86573| 0.38766|

371| 0.1046| 0.980| 0.0001023| 0.0000004| -1.20649| 0.65450| 1.86574| 0.38766|

372| 0.1046| 0.980| 0.0001003| 0.0000004| -1.20651| 0.65450| 1.86576| 0.38766|

373| 0.1046| 0.980| 0.0000983| 0.0000004| -1.20653| 0.65450| 1.86577| 0.38766|

Метод наискорейшего спуска

| Норма | Оценка |

Itr| Tau | q | невязки |погрешности| x[1] | x[2] | x[3] | x[4] |

1| 0.1802| 0.721| 5.5500000| 0.2792793| 0.18018| 0.36036| 0.54054| 0.72072|

2| 0.0864| 0.228| 2.6969787| 0.5564352| 0.24480| 0.32766| 0.70483| 0.59770|

3| 0.1268| 1.163| 2.0265103| -0.0267199| 0.20799| 0.51866| 0.74330| 0.57897|

4| 0.0878| 0.564| 1.4419523| 0.0831975| 0.23818| 0.49709| 0.85111| 0.52115|

5| 0.1294| 1.044| 0.9834076| -0.0046967| 0.20195| 0.60960| 0.88390| 0.52141|

6| 0.0894| 0.638| 0.8404148| 0.0406710| 0.21142| 0.59179| 0.95574| 0.48713|

7| 0.1338| 0.976| 0.5345878| 0.0017506| 0.17254| 0.66187| 0.97986| 0.49054|

8| 0.0918| 0.739| 0.5471428| 0.0182790| 0.17077| 0.64418| 1.03166| 0.46738|

9| 0.1389| 0.910| 0.3237312| 0.0046382| 0.12913| 0.69135| 1.04867| 0.47258|

10| 0.0938| 0.870| 0.4012486| 0.0061529| 0.12183| 0.67206| 1.08969| 0.45498|

11| 0.1423| 1.054| 0.2759031| -0.0022344| 0.07858| 0.70675| 1.10121| 0.46173|

12| 0.0948| 0.809| 0.3248809| 0.0082440| 0.06892| 0.68596| 1.13621| 0.44700|

13| 0.1438| 1.240| 0.2638654| -0.0084022| 0.02552| 0.71377| 1.14402| 0.45474|

14| 0.0952| 0.723| 0.2803696| 0.0120323| 0.01504| 0.69226| 1.17540| 0.44164|

15| 0.1443| 1.358| 0.2499929| -0.0112229| -0.02756| 0.71612| 1.18095| 0.44982|

16| 0.0953| 0.679| 0.2508682| 0.0136639| -0.03815| 0.69460| 1.20988| 0.43778|

17| 0.1444| 1.427| 0.2354073| -0.0123493| -0.07944| 0.71600| 1.21412| 0.44604|

18| 0.0954| 0.657| 0.2290126| 0.0141751| -0.08985| 0.69491| 1.24122| 0.43476|

19| 0.1445| 1.466| 0.2209694| -0.0126321| -0.12959| 0.71463| 1.24469| 0.44289|

20| 0.0954| 0.645| 0.2114451| 0.0141219| -0.13968| 0.69419| 1.27031| 0.43224|

21| 0.1445| 1.488| 0.2071526| -0.0124976| -0.17779| 0.71266| 1.27332| 0.44014|

22| 0.0954| 0.638| 0.1965350| 0.0137908| -0.18750| 0.69298| 1.29764| 0.43002|

23| 0.1445| 1.500| 0.1941656| -0.0121609| -0.22398| 0.71042| 1.30036| 0.43763|

24| 0.0954| 0.635| 0.1834381| 0.0133290| -0.23330| 0.69155| 1.32351| 0.42800|

25| 0.1445| 1.507| 0.1820682| -0.0117313| -0.26818| 0.70811| 1.32602| 0.43530|

26| 0.0954| 0.633| 0.1716873| 0.0128120| -0.27710| 0.69004| 1.34809| 0.42613|

27| 0.1445| 1.510| 0.1708457| -0.0112647| -0.31044| 0.70580| 1.35044| 0.43312|

28| 0.0954| 0.632| 0.1610051| 0.0122786| -0.31896| 0.68852| 1.37150| 0.42436|

29| 0.1445| 1.512| 0.1604496| -0.0107897| -0.35081| 0.70354| 1.37372| 0.43105|

30| 0.0954| 0.631| 0.1512140| 0.0117485| -0.35895| 0.68703| 1.39382| 0.42269|

31| 0.1445| 1.513| 0.1508185| -0.0103207| -0.38937| 0.70136| 1.39593| 0.42908|

32| 0.0954| 0.631| 0.1421906| 0.0112314| -0.39715| 0.68558| 1.41512| 0.42110|

33| 0.1445| 1.514| 0.1418889| -0.0098648| -0.42621| 0.69926| 1.41712| 0.42721|

34| 0.0954| 0.631| 0.1338433| 0.0107319| -0.43364| 0.68419| 1.43545| 0.41959|

35| 0.1445| 1.514| 0.1336002| -0.0094251| -0.46139| 0.69725| 1.43736| 0.42542|

36| 0.0954| 0.631| 0.1260999| 0.0102518| -0.46849| 0.68286| 1.45485| 0.41815|

37| 0.1445| 1.515| 0.1258958| -0.0090031| -0.49499| 0.69533| 1.45668| 0.42372|

38| 0.0954| 0.631| 0.1189009| 0.0097919| -0.50176| 0.68158| 1.47339| 0.41677|

39| 0.1445| 1.515| 0.1187249| -0.0085989| -0.52707| 0.69349| 1.47513| 0.42209|

40| 0.0954| 0.630| 0.1121958| 0.0093518| -0.53355| 0.68036| 1.49108| 0.41546|

41| 0.1445| 1.515| 0.1120413| -0.0082124| -0.55771| 0.69173| 1.49274| 0.42054|

42| 0.0954| 0.630| 0.1059410| 0.0089311| -0.56389| 0.67920| 1.50798| 0.41421|

43| 0.1445| 1.515| 0.1058039| -0.0078429| -0.58697| 0.69005| 1.50957| 0.41906|

44| 0.0954| 0.630| 0.1000981| 0.0085292| -0.59288| 0.67808| 1.52412| 0.41301|

45| 0.1445| 1.515| 0.0999754| -0.0074899| -0.61492| 0.68845| 1.52563| 0.41764|

46| 0.0954| 0.630| 0.0946331| 0.0081452| -0.62056| 0.67702| 1.53953| 0.41186|

47| 0.1445| 1.515| 0.0945228| -0.0071527| -0.64160| 0.68692| 1.54097| 0.41629|

48| 0.0954| 0.630| 0.0895155| 0.0077785| -0.64699| 0.67600| 1.55424| 0.41077|

49| 0.1445| 1.515| 0.0894160| -0.0068306| -0.66709| 0.68546| 1.55563| 0.41500|

50| 0.0954| 0.630| 0.0847181| 0.0074282| -0.67223| 0.67503| 1.56830| 0.40973|

51| 0.1445| 1.515| 0.0846281| -0.0065230| -0.69142| 0.68406| 1.56962| 0.41376|

52| 0.0954| 0.630| 0.0802164| 0.0070937| -0.69633| 0.67411| 1.58172| 0.40873|

53| 0.1445| 1.515| 0.0801349| -0.0062293| -0.71466| 0.68273| 1.58298| 0.41259|

54| 0.0954| 0.630| 0.0759882| 0.0067742| -0.71935| 0.67322| 1.59453| 0.40778|

55| 0.1445| 1.515| 0.0759142| -0.0059488| -0.73686| 0.68146| 1.59574| 0.41146|

56| 0.0954| 0.630| 0.0720134| 0.0064692| -0.74133| 0.67238| 1.60677| 0.40687|

57| 0.1445| 1.515| 0.0719461| -0.0056809| -0.75805| 0.68024| 1.60792| 0.41039|

58| 0.0954| 0.630| 0.0682737| 0.0061779| -0.76233| 0.67157| 1.61846| 0.40601|

59| 0.1445| 1.515| 0.0682125| -0.0054250| -0.77829| 0.67908| 1.61956| 0.40936|

60| 0.0954| 0.630| 0.0647525| 0.0058996| -0.78237| 0.67080| 1.62962| 0.40518|

61| 0.1445| 1.515| 0.0646967| -0.0051807| -0.79762| 0.67797| 1.63067| 0.40838|

62| 0.0954| 0.630| 0.0614346| 0.0056340| -0.80152| 0.67006| 1.64028| 0.40439|

63| 0.1445| 1.515| 0.0613837| -0.0049474| -0.81608| 0.67691| 1.64128| 0.40745|

64| 0.0954| 0.630| 0.0583062| 0.0053802| -0.81980| 0.66936| 1.65046| 0.40363|

65| 0.1445| 1.515| 0.0582596| -0.0047246| -0.83370| 0.67590| 1.65141| 0.40656|

66| 0.0954| 0.630| 0.0553544| 0.0051379| -0.83726| 0.66869| 1.66018| 0.40291|

67| 0.1445| 1.515| 0.0553118| -0.0045118| -0.85054| 0.67493| 1.66109| 0.40570|

68| 0.0954| 0.630| 0.0525676| 0.0049066| -0.85393| 0.66805| 1.66946| 0.40222|

69| 0.1445| 1.515| 0.0525287| -0.0043087| -0.86661| 0.67401| 1.67033| 0.40489|

70| 0.0954| 0.630| 0.0499351| 0.0046856| -0.86985| 0.66743| 1.67833| 0.40156|

71| 0.1445| 1.515| 0.0498994| -0.0041146| -0.88196| 0.67313| 1.67916| 0.40411|

72| 0.0954| 0.630| 0.0474470| 0.0044746| -0.88506| 0.66685| 1.68679| 0.40094|

73| 0.1445| 1.515| 0.0474143| -0.0039293| -0.89662| 0.67229| 1.68759| 0.40337|

74| 0.0954| 0.630| 0.0450942| 0.0042731| -0.89958| 0.66629| 1.69488| 0.40034|

75| 0.1445| 1.515| 0.0450641| -0.0037524| -0.91062| 0.67149| 1.69563| 0.40266|

76| 0.0954| 0.630| 0.0428681| 0.0040806| -0.91344| 0.66576| 1.70260| 0.39976|

77| 0.1445| 1.515| 0.0428405| -0.0035834| -0.92399| 0.67072| 1.70332| 0.40198|

78| 0.0954| 0.630| 0.0407611| 0.0038969| -0.92668| 0.66525| 1.70997| 0.39922|

79| 0.1445| 1.515| 0.0407357| -0.0034220| -0.93675| 0.66999| 1.71066| 0.40133|

80| 0.0954| 0.630| 0.0387659| 0.0037214| -0.93933| 0.66476| 1.71701| 0.39869|

81| 0.1445| 1.515| 0.0387424| -0.0032679| -0.94895| 0.66929| 1.71767| 0.40072|

82| 0.0954| 0.630| 0.0368757| 0.0035538| -0.95141| 0.66430| 1.72373| 0.39820|

83| 0.1445| 1.515| 0.0368541| -0.0031207| -0.96059| 0.66862| 1.72436| 0.40013|

84| 0.0954| 0.630| 0.0350843| 0.0033937| -0.96294| 0.66386| 1.73015| 0.39772|

85| 0.1445| 1.515| 0.0350644| -0.0029802| -0.97171| 0.66798| 1.73075| 0.39956|

86| 0.0954| 0.630| 0.0333860| 0.0032409| -0.97395| 0.66343| 1.73628| 0.39726|

87| 0.1445| 1.515| 0.0333676| -0.0028460| -0.98233| 0.66737| 1.73686| 0.39902|

88| 0.0954| 0.630| 0.0317754| 0.0030950| -0.98447| 0.66303| 1.74214| 0.39683|

89| 0.1445| 1.515| 0.0317584| -0.0027178| -0.99246| 0.66679| 1.74269| 0.39851|

90| 0.0954| 0.630| 0.0302473| 0.0029556| -0.99451| 0.66264| 1.74773| 0.39641|

91| 0.1445| 1.515| 0.0302316| -0.0025954| -1.00215| 0.66623| 1.74825| 0.39802|

92| 0.0954| 0.630| 0.0287972| 0.0028225| -1.00410| 0.66227| 1.75307| 0.39602|

93| 0.1445| 1.515| 0.0287826| -0.0024785| -1.01139| 0.66570| 1.75357| 0.39755|

94| 0.0954| 0.630| 0.0274206| 0.0026954| -1.01326| 0.66192| 1.75817| 0.39564|

95| 0.1445| 1.515| 0.0274071| -0.0023669| -1.02022| 0.66520| 1.75865| 0.39710|

96| 0.0954| 0.630| 0.0261134| 0.0025740| -1.02201| 0.66158| 1.76304| 0.39528|

97| 0.1445| 1.515| 0.0261009| -0.0022603| -1.02866| 0.66471| 1.76350| 0.39668|

98| 0.0954| 0.630| 0.0248718| 0.0024581| -1.03036| 0.66126| 1.76769| 0.39493|

99| 0.1445| 1.515| 0.0248602| -0.0021585| -1.03671| 0.66425| 1.76813| 0.39627|

100| 0.0954| 0.630| 0.0236922| 0.0023474| -1.03834| 0.66096| 1.77213| 0.39460|

101| 0.1445| 1.515| 0.0236814| -0.0020613| -1.04440| 0.66381| 1.77255| 0.39588|

102| 0.0954| 0.630| 0.0225712| 0.0022417| -1.04595| 0.66066| 1.77637| 0.39429|

103| 0.1445| 1.515| 0.0225612| -0.0019685| -1.05175| 0.66339| 1.77677| 0.39551|

104| 0.0954| 0.630| 0.0215057| 0.0021407| -1.05323| 0.66038| 1.78042| 0.39399|

105| 0.1445| 1.515| 0.0214964| -0.0018798| -1.05876| 0.66299| 1.78080| 0.39515|

106| 0.0954| 0.630| 0.0204927| 0.0020443| -1.06017| 0.66012| 1.78429| 0.39370|

107| 0.1445| 1.515| 0.0204840| -0.0017952| -1.06546| 0.66260| 1.78465| 0.39481|

108| 0.0954| 0.630| 0.0195294| 0.0019522| -1.06681| 0.65986| 1.78798| 0.39343|

109| 0.1445| 1.515| 0.0195213| -0.0017143| -1.07185| 0.66224| 1.78833| 0.39449|

110| 0.0954| 0.630| 0.0186132| 0.0018643| -1.07314| 0.65962| 1.79151| 0.39317|

111| 0.1445| 1.515| 0.0186057| -0.0016371| -1.07796| 0.66188| 1.79184| 0.39418|

112| 0.0954| 0.630| 0.0177416| 0.0017804| -1.07919| 0.65939| 1.79488| 0.39292|

113| 0.1445| 1.515| 0.0177346| -0.0015634| -1.08379| 0.66155| 1.79519| 0.39388|

114| 0.0954| 0.630| 0.0169123| 0.0017002| -1.08497| 0.65916| 1.79809| 0.39268|

115| 0.1445| 1.515| 0.0169058| -0.0014930| -1.08936| 0.66123| 1.79839| 0.39360|

116| 0.0954| 0.630| 0.0161231| 0.0016236| -1.09049| 0.65895| 1.80116| 0.39245|

117| 0.1445| 1.515| 0.0161170| -0.0014258| -1.09468| 0.66093| 1.80145| 0.39333|

118| 0.0954| 0.630| 0.0153720| 0.0015505| -1.09575| 0.65875| 1.80410| 0.39223|

119| 0.1445| 1.515| 0.0153663| -0.0013616| -1.09976| 0.66063| 1.80437| 0.39307|

120| 0.0954| 0.630| 0.0146570| 0.0014807| -1.10079| 0.65856| 1.80690| 0.39202|

121| 0.1445| 1.515| 0.0146516| -0.0013002| -1.10461| 0.66036| 1.80716| 0.39283|

122| 0.0954| 0.630| 0.0139762| 0.0014140| -1.10559| 0.65837| 1.80957| 0.39183|

123| 0.1445| 1.515| 0.0139712| -0.0012417| -1.10924| 0.66009| 1.80983| 0.39259|

124| 0.0954| 0.630| 0.0133280| 0.0013503| -1.11018| 0.65819| 1.81213| 0.39164|

125| 0.1445| 1.515| 0.0133233| -0.0011858| -1.11367| 0.65984| 1.81237| 0.39237|

126| 0.0954| 0.630| 0.0127107| 0.0012895| -1.11456| 0.65803| 1.81457| 0.39146|

127| 0.1445| 1.515| 0.0127063| -0.0011324| -1.11789| 0.65959| 1.81480| 0.39216|

128| 0.0954| 0.630| 0.0121227| 0.0012314| -1.11875| 0.65787| 1.81690| 0.39128|

129| 0.1445| 1.515| 0.0121186| -0.0010814| -1.12193| 0.65936| 1.81712| 0.39195|

130| 0.0954| 0.630| 0.0115626| 0.0011760| -1.12274| 0.65771| 1.81912| 0.39112|

131| 0.1445| 1.515| 0.0115587| -0.0010327| -1.12578| 0.65914| 1.81933| 0.39176|

132| 0.0954| 0.630| 0.0110290| 0.0011230| -1.12656| 0.65756| 1.82125| 0.39096|

133| 0.1445| 1.515| 0.0110254| -0.0009862| -1.12946| 0.65893| 1.82145| 0.39157|

134| 0.0954| 0.630| 0.0105206| 0.0010724| -1.13020| 0.65742| 1.82328| 0.39081|

135| 0.1445| 1.515| 0.0105172| -0.0009418| -1.13297| 0.65873| 1.82347| 0.39139|

136| 0.0954| 0.630| 0.0100362| 0.0010242| -1.13368| 0.65729| 1.82521| 0.39067|

137| 0.1445| 1.515| 0.0100330| -0.0008993| -1.13633| 0.65854| 1.82540| 0.39122|

138| 0.0954| 0.630| 0.0095745| 0.0009780| -1.13700| 0.65716| 1.82706| 0.39053|

139| 0.1445| 1.515| 0.0095715| -0.0008588| -1.13953| 0.65835| 1.82724| 0.39106|

140| 0.0954| 0.630| 0.0091345| 0.0009340| -1.14018| 0.65704| 1.82883| 0.39040|

141| 0.1445| 1.515| 0.0091316| -0.0008202| -1.14259| 0.65818| 1.82900| 0.39090|

142| 0.0954| 0.630| 0.0087151| 0.0008919| -1.14321| 0.65692| 1.83052| 0.39027|

143| 0.1445| 1.515| 0.0087124| -0.0007832| -1.14551| 0.65801| 1.83068| 0.39076|

144| 0.0954| 0.630| 0.0083153| 0.0008518| -1.14610| 0.65681| 1.83213| 0.39015|

145| 0.1445| 1.515| 0.0083128| -0.0007480| -1.14830| 0.65785| 1.83228| 0.39061|

146| 0.0954| 0.630| 0.0079341| 0.0008134| -1.14887| 0.65671| 1.83367| 0.39004|

147| 0.1445| 1.515| 0.0079318| -0.0007143| -1.15097| 0.65770| 1.83381| 0.39048|

148| 0.0954| 0.630| 0.0075708| 0.0007768| -1.15151| 0.65661| 1.83514| 0.38993|

149| 0.1445| 1.515| 0.0075685| -0.0006821| -1.15351| 0.65755| 1.83528| 0.39035|

150| 0.0954| 0.630| 0.0072243| 0.0007418| -1.15403| 0.65651| 1.83654| 0.38982|

151| 0.1445| 1.515| 0.0072222| -0.0006514| -1.15594| 0.65741| 1.83667| 0.39023|

152| 0.0954| 0.630| 0.0068939| 0.0007084| -1.15643| 0.65642| 1.83788| 0.38973|

153| 0.1445| 1.515| 0.0068919| -0.0006221| -1.15827| 0.65728| 1.83801| 0.39011|

154| 0.0954| 0.630| 0.0065789| 0.0006765| -1.15873| 0.65633| 1.83916| 0.38963|

155| 0.1445| 1.515| 0.0065770| -0.0005940| -1.16048| 0.65715| 1.83928| 0.39000|

156| 0.0954| 0.630| 0.0062784| 0.0006460| -1.16093| 0.65624| 1.84038| 0.38954|

157| 0.1445| 1.515| 0.0062767| -0.0005673| -1.16260| 0.65703| 1.84050| 0.38989|

158| 0.0954| 0.630| 0.0059919| 0.0006169| -1.16302| 0.65616| 1.84155| 0.38945|

159| 0.1445| 1.515| 0.0059902| -0.0005417| -1.16462| 0.65691| 1.84166| 0.38979|

160| 0.0954| 0.630| 0.0057186| 0.0005891| -1.16503| 0.65609| 1.84267| 0.38937|

161| 0.1445| 1.515| 0.0057170| -0.0005173| -1.16655| 0.65680| 1.84277| 0.38969|

162| 0.0954| 0.630| 0.0054579| 0.0005626| -1.16694| 0.65601| 1.84373| 0.38929|

163| 0.1445| 1.515| 0.0054564| -0.0004940| -1.16839| 0.65670| 1.84383| 0.38960|

164| 0.0954| 0.630| 0.0052092| 0.0005373| -1.16876| 0.65594| 1.84475| 0.38922|

165| 0.1445| 1.515| 0.0052078| -0.0004718| -1.17015| 0.65660| 1.84484| 0.38951|

166| 0.0954| 0.630| 0.0049720| 0.0005131| -1.17051| 0.65587| 1.84572| 0.38914|

167| 0.1445| 1.515| 0.0049707| -0.0004506| -1.17183| 0.65650| 1.84581| 0.38942|

168| 0.0954| 0.630| 0.0047457| 0.0004900| -1.17217| 0.65581| 1.84664| 0.38907|

169| 0.1445| 1.515| 0.0047445| -0.0004303| -1.17344| 0.65641| 1.84673| 0.38934|

170| 0.0954| 0.630| 0.0045298| 0.0004679| -1.17376| 0.65575| 1.84753| 0.38901|

171| 0.1445| 1.515| 0.0045287| -0.0004109| -1.17497| 0.65632| 1.84761| 0.38926|

172| 0.0954| 0.630| 0.0043239| 0.0004468| -1.17528| 0.65569| 1.84837| 0.38895|

173| 0.1445| 1.515| 0.0043227| -0.0003924| -1.17644| 0.65623| 1.84845| 0.38919|

174| 0.0954| 0.630| 0.0041273| 0.0004267| -1.17673| 0.65564| 1.84918| 0.38889|

175| 0.1445| 1.515| 0.0041263| -0.0003747| -1.17783| 0.65615| 1.84926| 0.38912|

176| 0.0954| 0.630| 0.0039398| 0.0004075| -1.17812| 0.65558| 1.84995| 0.38883|

177| 0.1445| 1.515| 0.0039388| -0.0003578| -1.17917| 0.65608| 1.85003| 0.38905|

178| 0.0954| 0.630| 0.0037609| 0.0003891| -1.17944| 0.65553| 1.85069| 0.38877|

179| 0.1445| 1.515| 0.0037599| -0.0003417| -1.18044| 0.65600| 1.85076| 0.38899|

180| 0.0954| 0.630| 0.0035902| 0.0003716| -1.18070| 0.65548| 1.85139| 0.38872|

181| 0.1445| 1.515| 0.0035893| -0.0003263| -1.18166| 0.65593| 1.85146| 0.38892|

182| 0.0954| 0.630| 0.0034273| 0.0003549| -1.18191| 0.65544| 1.85206| 0.38867|

183| 0.1445| 1.515| 0.0034264| -0.0003116| -1.18282| 0.65587| 1.85213| 0.38887|

184| 0.0954| 0.630| 0.0032718| 0.0003389| -1.18306| 0.65539| 1.85270| 0.38863|

185| 0.1445| 1.515| 0.0032710| -0.0002976| -1.18393| 0.65580| 1.85276| 0.38881|

186| 0.0954| 0.630| 0.0031234| 0.0003236| -1.18416| 0.65535| 1.85332| 0.38858|

187| 0.1445| 1.515| 0.0031226| -0.0002842| -1.18499| 0.65574| 1.85337| 0.38876|

188| 0.0954| 0.630| 0.0029818| 0.0003091| -1.18521| 0.65531| 1.85390| 0.38854|

189| 0.1445| 1.515| 0.0029811| -0.0002714| -1.18601| 0.65569| 1.85396| 0.38870|

190| 0.0954| 0.630| 0.0028467| 0.0002951| -1.18621| 0.65527| 1.85446| 0.38849|

191| 0.1445| 1.515| 0.0028460| -0.0002592| -1.18697| 0.65563| 1.85451| 0.38866|

192| 0.0954| 0.630| 0.0027177| 0.0002819| -1.18717| 0.65523| 1.85499| 0.38846|

193| 0.1445| 1.515| 0.0027170| -0.0002475| -1.18790| 0.65558| 1.85504| 0.38861|

194| 0.0954| 0.630| 0.0025946| 0.0002692| -1.18808| 0.65520| 1.85550| 0.38842|

195| 0.1445| 1.515| 0.0025939| -0.0002364| -1.18878| 0.65553| 1.85555| 0.38856|

196| 0.0954| 0.630| 0.0024771| 0.0002570| -1.18896| 0.65517| 1.85599| 0.38838|

197| 0.1445| 1.515| 0.0024765| -0.0002257| -1.18962| 0.65548| 1.85603| 0.38852|

198| 0.0954| 0.630| 0.0023649| 0.0002455| -1.18979| 0.65513| 1.85645| 0.38835|

199| 0.1445| 1.515| 0.0023644| -0.0002156| -1.19042| 0.65543| 1.85650| 0.38848|

200| 0.0954| 0.630| 0.0022579| 0.0002344| -1.19059| 0.65510| 1.85690| 0.38831|

201| 0.1445| 1.515| 0.0022574| -0.0002058| -1.19119| 0.65539| 1.85694| 0.38844|

202| 0.0954| 0.630| 0.0021557| 0.0002239| -1.19135| 0.65507| 1.85732| 0.38828|

203| 0.1445| 1.515| 0.0021552| -0.0001966| -1.19193| 0.65535| 1.85736| 0.38840|

204| 0.0954| 0.630| 0.0020582| 0.0002138| -1.19207| 0.65505| 1.85772| 0.38825|

205| 0.1445| 1.515| 0.0020577| -0.0001877| -1.19263| 0.65531| 1.85776| 0.38837|

206| 0.0954| 0.630| 0.0019651| 0.0002041| -1.19277| 0.65502| 1.85811| 0.38822|

207| 0.1445| 1.515| 0.0019646| -0.0001793| -1.19330| 0.65527| 1.85815| 0.38833|

208| 0.0954| 0.630| 0.0018762| 0.0001950| -1.19343| 0.65499| 1.85848| 0.38820|

209| 0.1445| 1.515| 0.0018758| -0.0001712| -1.19393| 0.65523| 1.85851| 0.38830|

210| 0.0954| 0.630| 0.0017914| 0.0001862| -1.19406| 0.65497| 1.85883| 0.38817|

211| 0.1445| 1.515| 0.0017910| -0.0001635| -1.19454| 0.65520| 1.85886| 0.38827|

212| 0.0954| 0.630| 0.0017104| 0.0001778| -1.19467| 0.65495| 1.85917| 0.38815|

213| 0.1445| 1.515| 0.0017100| -0.0001561| -1.19513| 0.65516| 1.85920| 0.38824|

214| 0.0954| 0.630| 0.0016331| 0.0001698| -1.19524| 0.65492| 1.85949| 0.38812|

215| 0.1445| 1.515| 0.0016327| -0.0001491| -1.19568| 0.65513| 1.85952| 0.38821|

216| 0.0954| 0.630| 0.0015593| 0.0001621| -1.19579| 0.65490| 1.85980| 0.38810|

217| 0.1445| 1.515| 0.0015589| -0.0001424| -1.19621| 0.65510| 1.85982| 0.38819|

218| 0.0954| 0.630| 0.0014888| 0.0001548| -1.19632| 0.65488| 1.86009| 0.38808|

219| 0.1445| 1.515| 0.0014885| -0.0001360| -1.19672| 0.65507| 1.86012| 0.38816|

220| 0.0954| 0.630| 0.0014216| 0.0001479| -1.19682| 0.65486| 1.86037| 0.38806|

221| 0.1445| 1.515| 0.0014213| -0.0001298| -1.19721| 0.65504| 1.86039| 0.38814|

222| 0.0954| 0.630| 0.0013574| 0.0001412| -1.19730| 0.65484| 1.86064| 0.38804|

223| 0.1445| 1.515| 0.0013571| -0.0001240| -1.19767| 0.65502| 1.86066| 0.38811|

224| 0.0954| 0.630| 0.0012961| 0.0001348| -1.19776| 0.65483| 1.86089| 0.38802|

225| 0.1445| 1.515| 0.0012958| -0.0001184| -1.19811| 0.65499| 1.86091| 0.38809|

226| 0.0954| 0.630| 0.0012375| 0.0001288| -1.19820| 0.65481| 1.86113| 0.38800|

227| 0.1445| 1.515| 0.0012372| -0.0001131| -1.19853| 0.65497| 1.86116| 0.38807|

228| 0.0954| 0.630| 0.0011817| 0.0001230| -1.19862| 0.65479| 1.86137| 0.38798|

229| 0.1445| 1.515| 0.0011814| -0.0001080| -1.19893| 0.65494| 1.86139| 0.38805|

230| 0.0954| 0.630| 0.0011283| 0.0001174| -1.19902| 0.65478| 1.86159| 0.38797|

231| 0.1445| 1.515| 0.0011280| -0.0001031| -1.19932| 0.65492| 1.86161| 0.38803|

232| 0.0954| 0.630| 0.0010774| 0.0001121| -1.19940| 0.65476| 1.86180| 0.38795|

233| 0.1445| 1.515| 0.0010771| -0.0000985| -1.19969| 0.65490| 1.86182| 0.38801|

234| 0.0954| 0.630| 0.0010287| 0.0001071| -1.19976| 0.65475| 1.86200| 0.38794|

235| 0.1445| 1.515| 0.0010285| -0.0000940| -1.20004| 0.65488| 1.86202| 0.38799|

236| 0.0954| 0.630| 0.0009823| 0.0001023| -1.20011| 0.65474| 1.86220| 0.38792|

237| 0.1445| 1.515| 0.0009821| -0.0000898| -1.20037| 0.65486| 1.86222| 0.38798|

238| 0.0954| 0.630| 0.0009380| 0.0000977| -1.20044| 0.65472| 1.86238| 0.38791|

239| 0.1445| 1.515| 0.0009378| -0.0000858| -1.20069| 0.65484| 1.86240| 0.38796|

240| 0.0954| 0.630| 0.0008957| 0.0000933| -1.20076| 0.65471| 1.86256| 0.38789|

241| 0.1445| 1.515| 0.0008954| -0.0000819| -1.20100| 0.65483| 1.86258| 0.38794|

242| 0.0954| 0.630| 0.0008552| 0.0000891| -1.20106| 0.65470| 1.86273| 0.38788|

243| 0.1445| 1.515| 0.0008550| -0.0000782| -1.20129| 0.65481| 1.86274| 0.38793|

244| 0.0954| 0.630| 0.0008167| 0.0000851| -1.20135| 0.65469| 1.86289| 0.38787|

245| 0.1445| 1.515| 0.0008165| -0.0000747| -1.20157| 0.65479| 1.86290| 0.38792|

246| 0.0954| 0.630| 0.0007798| 0.0000812| -1.20162| 0.65468| 1.86304| 0.38786|

247| 0.1445| 1.515| 0.0007796| -0.0000713| -1.20183| 0.65478| 1.86306| 0.38790|

248| 0.0954| 0.630| 0.0007446| 0.0000776| -1.20189| 0.65467| 1.86319| 0.38785|

249| 0.1445| 1.515| 0.0007445| -0.0000681| -1.20209| 0.65476| 1.86320| 0.38789|

250| 0.0954| 0.630| 0.0007110| 0.0000741| -1.20214| 0.65466| 1.86333| 0.38784|

251| 0.1445| 1.515| 0.0007109| -0.0000650| -1.20233| 0.65475| 1.86334| 0.38788|

252| 0.0954| 0.630| 0.0006790| 0.0000707| -1.20238| 0.65465| 1.86346| 0.38783|

253| 0.1445| 1.515| 0.0006788| -0.0000621| -1.20256| 0.65474| 1.86348| 0.38787|

254| 0.0954| 0.630| 0.0006484| 0.0000676| -1.20261| 0.65464| 1.86359| 0.38782|

255| 0.1445| 1.515| 0.0006482| -0.0000593| -1.20278| 0.65472| 1.86360| 0.38785|

256| 0.0954| 0.630| 0.0006191| 0.0000645| -1.20283| 0.65463| 1.86371| 0.38781|

257| 0.1445| 1.515| 0.0006190| -0.0000566| -1.20300| 0.65471| 1.86372| 0.38784|

258| 0.0954| 0.630| 0.0005912| 0.0000616| -1.20304| 0.65462| 1.86383| 0.38780|

259| 0.1445| 1.515| 0.0005911| -0.0000541| -1.20320| 0.65470| 1.86384| 0.38783|

260| 0.0954| 0.630| 0.0005645| 0.0000588| -1.20324| 0.65462| 1.86394| 0.38779|

261| 0.1445| 1.515| 0.0005644| -0.0000517| -1.20339| 0.65469| 1.86395| 0.38782|

262| 0.0954| 0.630| 0.0005391| 0.0000562| -1.20343| 0.65461| 1.86405| 0.38778|

263| 0.1445| 1.515| 0.0005390| -0.0000493| -1.20357| 0.65468| 1.86406| 0.38781|

264| 0.0954| 0.630| 0.0005148| 0.0000537| -1.20361| 0.65460| 1.86415| 0.38778|

265| 0.1445| 1.515| 0.0005147| -0.0000471| -1.20375| 0.65467| 1.86416| 0.38781|

266| 0.0954| 0.630| 0.0004916| 0.0000512| -1.20379| 0.65460| 1.86425| 0.38777|

267| 0.1445| 1.515| 0.0004915| -0.0000450| -1.20392| 0.65466| 1.86425| 0.38780|

268| 0.0954| 0.630| 0.0004694| 0.0000489| -1.20395| 0.65459| 1.86434| 0.38776|

269| 0.1445| 1.515| 0.0004693| -0.0000430| -1.20408| 0.65465| 1.86435| 0.38779|

270| 0.0954| 0.630| 0.0004483| 0.0000467| -1.20411| 0.65458| 1.86443| 0.38776|

271| 0.1445| 1.515| 0.0004482| -0.0000410| -1.20423| 0.65464| 1.86443| 0.38778|

272| 0.0954| 0.630| 0.0004280| 0.0000446| -1.20426| 0.65458| 1.86451| 0.38775|

273| 0.1445| 1.515| 0.0004280| -0.0000392| -1.20438| 0.65463| 1.86452| 0.38777|

274| 0.0954| 0.630| 0.0004088| 0.0000426| -1.20441| 0.65457| 1.86459| 0.38774|

275| 0.1445| 1.515| 0.0004087| -0.0000374| -1.20452| 0.65462| 1.86460| 0.38777|

276| 0.0954| 0.630| 0.0003903| 0.0000407| -1.20455| 0.65457| 1.86467| 0.38774|

277| 0.1445| 1.515| 0.0003902| -0.0000357| -1.20465| 0.65462| 1.86468| 0.38776|

278| 0.0954| 0.630| 0.0003727| 0.0000389| -1.20468| 0.65456| 1.86474| 0.38773|

279| 0.1445| 1.515| 0.0003727| -0.0000341| -1.20478| 0.65461| 1.86475| 0.38775|

280| 0.0954| 0.630| 0.0003559| 0.0000371| -1.20480| 0.65456| 1.86481| 0.38773|

281| 0.1445| 1.515| 0.0003559| -0.0000326| -1.20490| 0.65460| 1.86482| 0.38775|

282| 0.0954| 0.630| 0.0003399| 0.0000354| -1.20492| 0.65455| 1.86488| 0.38772|

283| 0.1445| 1.515| 0.0003398| -0.0000311| -1.20502| 0.65459| 1.86489| 0.38774|

284| 0.0954| 0.630| 0.0003246| 0.0000338| -1.20504| 0.65455| 1.86494| 0.38772|

285| 0.1445| 1.515| 0.0003245| -0.0000297| -1.20513| 0.65459| 1.86495| 0.38774|

286| 0.0954| 0.630| 0.0003099| 0.0000323| -1.20515| 0.65454| 1.86500| 0.38771|

287| 0.1445| 1.515| 0.0003099| -0.0000284| -1.20523| 0.65458| 1.86501| 0.38773|

288| 0.0954| 0.630| 0.0002960| 0.0000309| -1.20525| 0.65454| 1.86506| 0.38771|

289| 0.1445| 1.515| 0.0002959| -0.0000271| -1.20533| 0.65458| 1.86507| 0.38772|

290| 0.0954| 0.630| 0.0002826| 0.0000295| -1.20535| 0.65454| 1.86512| 0.38770|

291| 0.1445| 1.515| 0.0002826| -0.0000259| -1.20543| 0.65457| 1.86512| 0.38772|

292| 0.0954| 0.630| 0.0002699| 0.0000281| -1.20545| 0.65453| 1.86517| 0.38770|

293| 0.1445| 1.515| 0.0002698| -0.0000247| -1.20552| 0.65457| 1.86518| 0.38772|

294| 0.0954| 0.630| 0.0002577| 0.0000269| -1.20554| 0.65453| 1.86522| 0.38770|

295| 0.1445| 1.515| 0.0002577| -0.0000236| -1.20561| 0.65456| 1.86523| 0.38771|

296| 0.0954| 0.630| 0.0002461| 0.0000257| -1.20563| 0.65452| 1.86527| 0.38769|

297| 0.1445| 1.515| 0.0002461| -0.0000225| -1.20569| 0.65456| 1.86528| 0.38771|

298| 0.0954| 0.630| 0.0002350| 0.0000245| -1.20571| 0.65452| 1.86532| 0.38769|

299| 0.1445| 1.515| 0.0002350| -0.0000215| -1.20577| 0.65455| 1.86532| 0.38770|

300| 0.0954| 0.630| 0.0002245| 0.0000234| -1.20579| 0.65452| 1.86536| 0.38769|

301| 0.1445| 1.515| 0.0002244| -0.0000206| -1.20585| 0.65455| 1.86537| 0.38770|

302| 0.0954| 0.630| 0.0002143| 0.0000224| -1.20587| 0.65452| 1.86540| 0.38768|

303| 0.1445| 1.515| 0.0002143| -0.0000196| -1.20592| 0.65454| 1.86541| 0.38769|

304| 0.0954| 0.630| 0.0002047| 0.0000213| -1.20594| 0.65451| 1.86544| 0.38768|

305| 0.1445| 1.515| 0.0002046| -0.0000187| -1.20599| 0.65454| 1.86545| 0.38769|

306| 0.0954| 0.630| 0.0001955| 0.0000204| -1.20601| 0.65451| 1.86548| 0.38768|

307| 0.1445| 1.515| 0.0001954| -0.0000179| -1.20606| 0.65453| 1.86549| 0.38769|

308| 0.0954| 0.630| 0.0001867| 0.0000195| -1.20607| 0.65451| 1.86552| 0.38767|

309| 0.1445| 1.515| 0.0001866| -0.0000171| -1.20613| 0.65453| 1.86552| 0.38768|

310| 0.0954| 0.630| 0.0001782| 0.0000186| -1.20614| 0.65451| 1.86555| 0.38767|

311| 0.1445| 1.515| 0.0001782| -0.0000163| -1.20619| 0.65453| 1.86556| 0.38768|

312| 0.0954| 0.630| 0.0001702| 0.0000178| -1.20620| 0.65450| 1.86559| 0.38767|

313| 0.1445| 1.515| 0.0001702| -0.0000156| -1.20624| 0.65452| 1.86559| 0.38768|

314| 0.0954| 0.630| 0.0001625| 0.0000170| -1.20626| 0.65450| 1.86562| 0.38767|

315| 0.1445| 1.515| 0.0001625| -0.0000149| -1.20630| 0.65452| 1.86562| 0.38768|

316| 0.0954| 0.630| 0.0001552| 0.0000162| -1.20631| 0.65450| 1.86565| 0.38766|

317| 0.1445| 1.515| 0.0001552| -0.0000142| -1.20635| 0.65452| 1.86565| 0.38767|

318| 0.0954| 0.630| 0.0001482| 0.0000155| -1.20636| 0.65450| 1.86568| 0.38766|

319| 0.1445| 1.515| 0.0001482| -0.0000136| -1.20640| 0.65452| 1.86568| 0.38767|

320| 0.0954| 0.630| 0.0001416| 0.0000148| -1.20641| 0.65449| 1.86571| 0.38766|

321| 0.1445| 1.515| 0.0001415| -0.0000130| -1.20645| 0.65451| 1.86571| 0.38767|

322| 0.0954| 0.630| 0.0001352| 0.0000141| -1.20646| 0.65449| 1.86573| 0.38766|

323| 0.1445| 1.515| 0.0001351| -0.0000124| -1.20650| 0.65451| 1.86574| 0.38767|

324| 0.0954| 0.630| 0.0001291| 0.0000135| -1.20651| 0.65449| 1.86576| 0.38766|

325| 0.1445| 1.515| 0.0001291| -0.0000118| -1.20654| 0.65451| 1.86576| 0.38766|

326| 0.0954| 0.630| 0.0001233| 0.0000129| -1.20655| 0.65449| 1.86578| 0.38765|

327| 0.1445| 1.515| 0.0001232| -0.0000113| -1.20658| 0.65450| 1.86579| 0.38766|

328| 0.0954| 0.630| 0.0001177| 0.0000123| -1.20659| 0.65449| 1.86581| 0.38765|

329| 0.1445| 1.515| 0.0001177| -0.0000108| -1.20662| 0.65450| 1.86581| 0.38766|

330| 0.0954| 0.630| 0.0001124| 0.0000117| -1.20663| 0.65449| 1.86583| 0.38765|

331| 0.1445| 1.515| 0.0001124| -0.0000103| -1.20666| 0.65450| 1.86583| 0.38766|

332| 0.0954| 0.630| 0.0001074| 0.0000112| -1.20667| 0.65448| 1.86585| 0.38765|

333| 0.1445| 1.515| 0.0001073| -0.0000098| -1.20670| 0.65450| 1.86585| 0.38766|

334| 0.0954| 0.630| 0.0001025| 0.0000107| -1.20671| 0.65448| 1.86587| 0.38765|

335| 0.1445| 1.515| 0.0001025| -0.0000094| -1.20673| 0.65450| 1.86587| 0.38765|

336| 0.0954| 0.630| 0.0000979| 0.0000102| -1.20674| 0.65448| 1.86589| 0.38765|

Метод ПВР - выбор оптимального w

w = 0.1 Itr = 531

w = 0.2 Itr = 247

w = 0.3 Itr = 152

w = 0.4 Itr = 108

w = 0.5 Itr = 83

w = 0.6 Itr = 66

w = 0.7 Itr = 54

w = 0.8 Itr = 47

w = 0.9 Itr = 40

w = 1.0 Itr = 35

w = 1.1 Itr = 29

w = 1.2 Itr = 24

w = 1.3 Itr = 18

w = 1.4 Itr = 14

w = 1.5 Itr = 21

w = 1.6 Itr = 19

w = 1.7 Itr = 26

w = 1.8 Itr = 41

w = 1.9 Itr = 71

w\* = 1.4 ItrMin = 14

Метод ПВР

| Норма | Оценка |

Itr| Tau | q | невязки |погрешности| x[1] | x[2] | x[3] | x[4] |

1| 1.4000| 0.925| 4.3257155| 0.0752975| 0.56000| 0.57584| 0.92470| 0.77222|

2| 1.4000| 0.463| 1.5556315| 0.4967884| 0.13209| 0.46953| 1.00699| 0.45445|

3| 1.4000| 1.012| 1.4289505| -0.0052402| -0.30107| 0.50220| 1.28326| 0.52798|

4| 1.4000| 0.651| 0.8098631| 0.1510875| -0.58314| 0.52107| 1.45344| 0.46279|

5| 1.4000| 0.867| 0.6877394| 0.0375906| -0.82761| 0.54930| 1.60555| 0.45108|

6| 1.4000| 0.682| 0.4652573| 0.0777141| -0.99437| 0.57470| 1.71213| 0.42661|

7| 1.4000| 0.707| 0.3340068| 0.0487963| -1.11234| 0.59748| 1.78917| 0.41296|

8| 1.4000| 0.631| 0.2205425| 0.0435674| -1.18674| 0.61592| 1.83941| 0.40133|

9| 1.4000| 0.591| 0.1409813| 0.0304578| -1.23068| 0.63022| 1.87037| 0.39393|

10| 1.4000| 0.499| 0.0821265| 0.0220233| -1.25260| 0.64067| 1.88710| 0.38899|

11| 1.4000| 0.346| 0.0418558| 0.0143403| -1.26017| 0.64791| 1.89429| 0.38616|

12| 1.4000| 0.621| 0.0151198| 0.0028687| -1.25892| 0.65262| 1.89538| 0.38471|

13| 1.4000| 1.278| 0.0056741| -0.0013067| -1.25290| 0.65543| 1.89299| 0.38421|

14| 1.4000| 1.338| 0.0102948| -0.0020357| -1.24485| 0.65690| 1.88889| 0.38429|

15| 1.4000| 1.037| 0.0142855| -0.0002970| -1.23651| 0.65746| 1.88427| 0.38467|

16| 1.4000| 0.916| 0.0150519| 0.0006976| -1.22886| 0.65746| 1.87984| 0.38519|

17| 1.4000| 0.846| 0.0139298| 0.0011803| -1.22239| 0.65715| 1.87598| 0.38573|

18| 1.4000| 0.795| 0.0118668| 0.0013263| -1.21724| 0.65670| 1.87282| 0.38623|

19| 1.4000| 0.753| 0.0094897| 0.0012705| -1.21337| 0.65621| 1.87039| 0.38665|

20| 1.4000| 0.714| 0.0071830| 0.0011079| -1.21060| 0.65576| 1.86862| 0.38698|

21| 1.4000| 0.674| 0.0051542| 0.0009028| -1.20874| 0.65538| 1.86739| 0.38723|

22| 1.4000| 0.627| 0.0034906| 0.0006952| -1.20757| 0.65507| 1.86659| 0.38741|

23| 1.4000| 0.565| 0.0022028| 0.0005077| -1.20691| 0.65483| 1.86612| 0.38753|

24| 1.4000| 0.468| 0.0012578| 0.0003508| -1.20661| 0.65466| 1.86587| 0.38761|

25| 1.4000| 0.376| 0.0006016| 0.0001928| -1.20652| 0.65454| 1.86578| 0.38765|

26| 1.4000| 0.636| 0.0001741| 0.0000422| -1.20658| 0.65447| 1.86578| 0.38767|

27| 1.4000| 1.664| 0.0001176| -0.0000490| -1.20670| 0.65443| 1.86584| 0.38768|

28| 1.4000| 1.212| 0.0002158| -0.0000260| -1.20685| 0.65441| 1.86592| 0.38767|

29| 1.4000| 0.994| 0.0002680| 0.0000009| -1.20700| 0.65440| 1.86600| 0.38767|

30| 1.4000| 0.893| 0.0002694| 0.0000158| -1.20713| 0.65440| 1.86608| 0.38766|

31| 1.4000| 0.830| 0.0002422| 0.0000225| -1.20724| 0.65441| 1.86614| 0.38765|

32| 1.4000| 0.782| 0.0002020| 0.0000239| -1.20732| 0.65442| 1.86620| 0.38764|

33| 1.4000| 0.742| 0.0001588| 0.0000221| -1.20739| 0.65442| 1.86624| 0.38763|

34| 1.4000| 0.703| 0.0001183| 0.0000189| -1.20743| 0.65443| 1.86627| 0.38763|

35| 1.4000| 0.661| 0.0000835| 0.0000151| -1.20746| 0.65444| 1.86629| 0.38762|

Метод сопряженных градиентов

| Норма | Оценка |

Itr| Tau | q | невязки |погрешности| x[1] | x[2] | x[3] | x[4] |

1| 0.1802| 0.721| 5.5500000| 0.2792793| 0.18018| 0.36036| 0.54054| 0.72072| 1.0000000

2| 0.0864| 0.338| 2.4235114| 0.4769031| 0.27243| 0.36463| 0.78436| 0.66515| 1.1128393

3| 0.1217| 1.571| 1.3650944| -0.1391081| 0.26420| 0.74756| 1.06264| 0.44120| 2.1422502

4| 2.7836| 3.843| 0.1590646| -1.0887530| -1.20748| 0.65445| 1.86630| 0.38762| 1.7732694

5| 0.0598| 0.000| 0.0000000| 1.4716787| -1.20748| 0.65445| 1.86630| 0.38762| 1.0000000

Число обусловленности 90.6991193

Теоретическая оценка чиcла итераций

Метод простых итераций 417

Метод наискорейшего спуска 835

Метод ПВР 21

Метод сопряженных градиентов 47

Сравнение с LU разложением

Разница между LU разложением и методом простых итераций

-9.4711782269e-04 -4.3464184027e-05 5.3407352267e-04 -4.2613505582e-05

Разница между LU разложением и методом наискорейшего спуска

-7.3590029572e-04 -2.8293907619e-05 4.0971623053e-04 -3.0403000716e-05

Разница между LU разложением и методом ПВР

-1.5595119165e-05 1.4847735936e-05 1.4768482951e-05 -5.8611480603e-06

Разница между LU разложением и методом сопряженных градиентов

4.2188474936e-15 2.1094237468e-15 -2.4424906542e-15 -3.5527136788e-15

Краткие выводы

Мы разобрались с итерационными методами решения СЛАУ. Реализовав 4 метода, можно сделать вывод, что метод сопряженных градиентов требует меньше всего итераций, что является преимуществом в экономии вычислительных затрат. Однако стоит принять во внимание ограничение, накладываемое на исходную матрицу – она должна быть симметричной и положительно определенной.

Текст программы

#include <iostream>

#include <algorithm>

#include <math.h>

#include <fstream>

#include <iomanip>

#define M\_PI 3.14159265358979323846

using namespace std;

const double minValue = 0.00001;

const double eps = 0.0001;

//распечатка вектора

void printVector(double\* vector, int N, bool expFormat = false) {

printf(" ");

for (int i = 0; i < N; i++) {

if (expFormat) {

printf("%.10e ", vector[i]);

}

else {

printf("%.7f ", vector[i]);

}

}

printf("\n");

}

//распечатка матрицы

void printMatrix(double\*\* matrix, int N, bool expFormat = false) {

for (int i = 0; i < N; i++) {

printVector(matrix[i], N, expFormat);

}

printf("\n");

}

//создаем в памяти матрицу

void createMatrix(double\*\*\* pMatrix, int N) {

auto& matrix = \*pMatrix;

matrix = new double\* [N];

for (int i = 0; i < N; i++) {

matrix[i] = new double[N];

}

}

//заполняем матрицу нулями

void fillMatrixAsEmpty(double\*\* matrix, int N) {

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

matrix[i][j] = 0.0;

}

}

}

//заполняем матрицу как единичную

void fillMatrixAsE(double\*\* matrix, int N) {

fillMatrixAsEmpty(matrix, N);

for (int i = 0; i < N; i++) {

matrix[i][i] = 1.0;

}

}

//копируем значения одной матрицы в другую

void copyMatrixToMatrix(double\*\* srcMatrix, double\*\* dstMatrix, int N) {

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

dstMatrix[i][j] = srcMatrix[i][j];

}

}

}

//копируем значения одной матрицы в другую

void copyVectorToVector(double\* srcVector, double\* dstVector, int N) {

for (int i = 0; i < N; i++) {

dstVector[i] = srcVector[i];

}

}

//очистка вектора

void vectorClear(double\* A, int N)

{

for (int i = 0; i < N; i++)

A[i] = 0.0;

}

//умножение матрицы на вектор

void matrixMulVec(double\*\* A, double\* B, double\* C, int N)

{

vectorClear(C, N);

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

C[i] += A[i][j] \* B[j];

}

//умножение матрицы на матрицу

void matrixMul(double\*\* A, double\*\* B, double\*\* C, int N)

{

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

for (int k = 0; k < N; k++)

C[i][j] += A[i][k] \* B[k][j];

}

//умножение матрицы на скаляр

void matrixMulScalar(double\*\* A, double scalar, double\*\* C, int N)

{

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

C[i][j] = A[i][j] \* scalar;

}

//вычитание матрицы

void matrixSub(double\*\* A, double\*\* B, double\*\* C, int N, double scalar = 1.0)

{

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

C[i][j] = A[i][j] - B[i][j] \* scalar;

}

//получение матрицы PA

double\*\* matrixPA(double\*\* A, int\* P, int N)

{

double\*\* PA = new double\* [N];

for (int i = 0; i < N; i++) {

PA[i] = A[P[i]];

}

return PA;

}

//сложение векторов

void vectorAdd(double\* A, double\* B, double\* C, int N)

{

for (int i = 0; i < N; i++)

C[i] = A[i] + B[i];

}

//вычитание векторов

void vectorSub(double\* A, double\* B, double\* C, int N)

{

for (int i = 0; i < N; i++)

C[i] = A[i] - B[i];

}

//умножение вектора на скаляр

void vectorMulScalar(double\* A, double scalar, double\* C, int N)

{

for (int i = 0; i < N; i++)

C[i] = A[i] \* scalar;

}

//скалярное произведенеие векторов

double vectorMul(double\* v1, double\* v2, int N)

{

double sum = 0;

for (int i = 0; i < N; i++)

sum += v1[i] \* v2[i];

return sum;

}

//норма вектора

double vectorEuNorm(double\* vector, int N)

{

double sum = 0;

for (int i = 0; i < N; i++)

sum += pow(vector[i], 2);

return sqrt(sum);

}

double vectorNorm(double\* v, int N)

{

double max = abs(v[0]);

for (int i = 1; i < N; i++)

if (abs(v[i]) > max)

max = abs(v[i]);

return max;

}

//определить строку с главным элементом (который максимален в текущем столбце)

int defineRowIdxWithMainValue(double\*\* matrix, int k, int N) {

int m = k;

double maxValue = 0.0;

for (int i = k; i < N; i++) {

if (abs(matrix[i][k]) > maxValue) {

m = i;

maxValue = abs(matrix[m][k]);

}

}

return m;

}

//LU разложение

void LUdecomposition(double\*\* L, double\*\* U, int\* P, int& rank, double& sign, int N) {

//иницилизируем подстановку P

for (int i = 0; i < N; i++) {

P[i] = i;

}

printf("U = A:\n");

printMatrix(U, N);

printf("L:\n");

printMatrix(L, N);

for (int k = 0; k < N; k++) {

auto rowIdx = defineRowIdxWithMainValue(U, k, N);

if (k != rowIdx) {

//Смена строк

swap(U[k], U[rowIdx]);

swap(L[k], L[rowIdx]);

swap(P[k], P[rowIdx]);

sign \*= -1.0;

}

//главный элемент

double mainValue = U[k][k];

if (abs(mainValue) < minValue) {

//Определяем ранг матрицы

rank = k;

return;

}

//заполняем матрицу L

for (int i = k; i < N; i++) {

L[i][k] = U[i][k];

//printMatrix(L, N);

}

for (int j = k; j < N; j++) {

U[k][j] /= mainValue;

}

//заполняем матрицу U

for (int i = k + 1; i < N; i++) {

for (int j = k; j < N; j++) {

U[i][j] = U[i][j] - L[i][k] \* U[k][j];

//printMatrix(U, N);

}

}

printf("\nk = %i\nm = %i\nU[m][k] = %.7f\nU:\n", k, rowIdx, mainValue);

printMatrix(U, N);

printf("L:\n");

printMatrix(L, N);

}

}

//Решение уравнения Ly = Pb

void SolveLy(double\*\* triangleMatrix, double\* X, double\* B, int N) {

for (int i = 0; i < N; i++) {

X[i] = B[i] / triangleMatrix[i][i];

for (int j = 0; j < i; j++) {

X[i] -= X[j] \* triangleMatrix[i][j] / triangleMatrix[i][i];

}

}

}

//Решение уравнения Ux = y

void SolveUx(double\*\* triangleMatrix, double\* X, double\* B, int N) {

for (int i = N - 1; i >= 0; i--) {

X[i] = B[i];

for (int j = N - 1; j > i; j--) {

X[i] -= X[j] \* triangleMatrix[i][j];

}

}

}

//Решение уравнения Ax = b, то есть LUx = Pb

void SolveSOLE(double\*\* L, double\*\* U, double\* X, int\* P, double\* B, int N) {

//Ax = b (A = PLU)

//LUx = Pb (Ux = y)

//Ly = Pb

double\* vectorY = new double[N];

//"умножаем" вектор b на матрицу перестановок

double\* vectorPB = new double[N];

for (int i = 0; i < N; i++) {

vectorPB[i] = B[P[i]];

}

SolveLy(L, vectorY, vectorPB, N);

//printVector(Y, N);

SolveUx(U, X, vectorY, N);

}

void SolveBackwardMatrix(double\*\* L, double\*\* U, double\*\* X, int\* P, int N) {

//LUX = PE

double\* vectorX = new double[N];

double\* vectorE = new double[N];

for (int t = 0; t < N; t++) {

vectorE[t] = 0.0;

}

for (int i = 0; i < N; i++) {

//формируем вектор Ei

if (i != 0) {

vectorE[i - 1] = 0.0;

}

vectorE[i] = 1.0;

//получаем вектор-столбец X

SolveSOLE(L, U, vectorX, P, vectorE, N);

//записываем его в матрицу X

for (int t = 0; t < N; t++) {

X[t][i] = vectorX[t];

}

}

}

//транспонировать матрицу

void transpose(double\*\* matrix, int N) {

for (int i = 0; i < N; i++) {

for (int j = i; j < N; j++) {

swap(matrix[j][i], matrix[i][j]);

}

}

}

//////////////////////////////////////////////////////////////////////////////////////////////////

//Нахождение максимального элемента в матрице, находящегося не на диагонали

void searchMaxElemMatrix(double\*\* matrix, const int N, int& imax, int& jmax) {

double max = 0.0;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

if (i != j && abs(matrix[i][j]) > max)

{

max = abs(matrix[i][j]);

imax = i;

jmax = j;

}

}

}

}

//tan(2\*alpha) = 2\*a[i][j]/(a[i][i]-a[j][j])

double getAlpha(double\*\* matrix, int imax, int jmax) {

double alpha;

if (matrix[imax][imax] - matrix[jmax][jmax] == 0)

{

alpha = M\_PI / 4;

}

else

{

alpha = atan(2 \* matrix[imax][jmax] / (matrix[imax][imax] - matrix[jmax][jmax])) / 2;

}

return alpha;

}

//Является ли матрица диагональной

bool isMatrixDiagonal(double\*\* matrix, const int N) {

double kvSum = 0.0;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

if (i != j)

{

kvSum += matrix[i][j] \* matrix[i][j];

}

}

}

return kvSum < minValue \* minValue;

}

//Получение диагональной матрицы

double\*\* getNewDiagonalMatrixByRotation(double\*\* matrix, const int N) {

double\* vectorI = new double[N];

double\* vectorJ = new double[N];

double\*\* rotatedMatrix = nullptr;

createMatrix(&rotatedMatrix, N);

copyMatrixToMatrix(matrix, rotatedMatrix, N);

//вспомогательная матрица

double\*\* B = nullptr;

createMatrix(&B, N);

while (!isMatrixDiagonal(rotatedMatrix, N)) {

//printMatrix(rotatedMatrix, N);

int imax, jmax;

searchMaxElemMatrix(rotatedMatrix, N, imax, jmax);

double alpha = getAlpha(rotatedMatrix, imax, jmax);

double c = cos(alpha);

double s = sin(alpha);

//результат умножения матрицы A в k-ом состоянии на матрицу вращения справа

copyMatrixToMatrix(rotatedMatrix, B, N);

for (int m = 0; m < N; m++) {

B[m][imax] = c \* rotatedMatrix[m][imax] + s \* rotatedMatrix[m][jmax];

B[m][jmax] = -s \* rotatedMatrix[m][imax] + c \* rotatedMatrix[m][jmax];

}

//результат умножения матрицы B на матрицу вращения слева

for (int m = 0; m < N; m++) {

vectorI[m] = c \* B[imax][m] + s \* B[jmax][m];

vectorJ[m] = -s \* B[imax][m] + c \* B[jmax][m];

}

swap(B[imax], vectorI);

swap(B[jmax], vectorJ);

copyMatrixToMatrix(B, rotatedMatrix, N);

}

return rotatedMatrix;

}

//Получение максимального собственного значения

double getMaxEigenvalue(double\*\* matrix, const int N) {

double max = 0;

for (int i = 0; i < N; i++)

{

if (abs(matrix[i][i]) > max)

max = abs(matrix[i][i]);

}

return max;

}

//Вычисление евклидовой нормы матрицы

double computEuclidNorm(double\*\* A, double\*\* trA, const int N) {

double\*\* newMatrix = nullptr;

createMatrix(&newMatrix, N);

fillMatrixAsEmpty(newMatrix, N);

matrixMul(trA, A, newMatrix, N);

//printMatrix(newMatrix, N);

newMatrix = getNewDiagonalMatrixByRotation(newMatrix, N);

//printMatrix(newMatrix, N);

double eigenvalue = getMaxEigenvalue(newMatrix, N);

return sqrt(eigenvalue);

}

//чтение входных данных

void readInputData(const char\* filename, double\*\*\* pA, double\*\* pB, int\* pN) {

std::ifstream file(filename);

if (!file.is\_open())

throw std::exception();

int idx = 0;

double value;

file >> \*pN;

int N = \*pN;

createMatrix(pA, N);

\*pB = new double[N];

while (file >> value)

{

if (idx < N \* N) {

(\*pA)[idx / N][idx % N] = value;

}

else {

(\*pB)[idx % N] = value;

}

idx++;

}

}

void printIterHeader()

{

cout << setw(8) << " " << setw(8) << " " << setw(7) << " " << "|" << setw(12) << "Норма" << "|" << setw(12) << "Оценка" << "|\n";

cout << setw(7) << "Itr" << "|" << setw(7) << "Tau" << "|" << setw(7) << "q" << "|" << setw(12) << "невязки" << "|" << setw(12) << "погрешности" << "|";

cout << setw(12) << "x[1]" << "|" << setw(12) << "x[2]" << "|" << setw(12) << "x[3]" << "|" << setw(12) << "x[4]" << "|\n";

}

void printIterStep(int itr, double tau, double q, double norm, double err, double\* v, int N, bool isAlpha = false, double alpha = 0.0)

{

cout << setw(7) << itr << "|";

cout << setw(7) << setprecision(4) << fixed << tau << "|";

cout << setw(7) << setprecision(3) << fixed << q << "|";

cout << setw(12) << setprecision(7) << fixed << norm << "|";

cout << setw(12) << setprecision(7) << fixed << err << "|";

for (int i = 0; i < N; i++)

cout << setw(12) << setprecision(5) << fixed << v[i] << "|";

if (isAlpha)

cout << setw(12) << setprecision(7) << fixed << alpha;

cout << endl;

}

//вычислить норму вектора-невязки

double calcVectorRNorm(double\*\* A, double\* B, double\* X, double\* prevX, double\* tempVector, int N) {

//R = Ax

matrixMulVec(A, prevX, tempVector, N);

//R = b - Ax

vectorSub(B, tempVector, tempVector, N);

//|R|

return vectorNorm(tempVector, N) / vectorNorm(X, N);

}

//оценка нормы матрицы перехода q

double calcQ(double\* X, double\* prevX, double\* tempVector, double\* prevDelta, int N) {

vectorSub(X, prevX, tempVector, N);

auto delta = vectorNorm(tempVector, N);

auto q = delta / \*prevDelta;

\*prevDelta = delta;

return q;

}

//считаем оценку погрешности

double calcError(double\* X, double\* prevX, double\* tempVector, double q, int N)

{

vectorSub(X, prevX, tempVector, N);

return vectorNorm(tempVector, N) \* (1 - q) / q;

}

//метод простой итерации

void SimpleIterationMethod(double\*\* A, double normA, double\* B, double\* X, int N)

{

auto prevX = new double[N];

auto tempVector = new double[N];

int itr = 1;

double tau = 0.9 \* 2 / normA;

double normR = 0;

auto prevDelta = 1.0;

vectorClear(prevX, N);

printIterHeader();

do {

//R = Ax

matrixMulVec(A, prevX, tempVector, N);

//R = b - Ax

vectorSub(B, tempVector, tempVector, N);

//(b - Ax) \* tau

vectorMulScalar(tempVector, tau, tempVector, N);

//x(k+1) = x(k) + (b - Ax) \* tau

vectorAdd(prevX, tempVector, X, N);

//норма невязки

normR = calcVectorRNorm(A, B, X, prevX, tempVector, N);

//оценка нормы матрицы перехода q

auto q = calcQ(X, prevX, tempVector, &prevDelta, N);

//погрешность

auto err = calcError(X, prevX, tempVector, q, N);

//вывод

printIterStep(itr++, tau, q, normR, err, X, N);

copyVectorToVector(X, prevX, N);

} while (normR > eps);

}

//метод наискорейшего спуска

void FastDescentMethod(double\*\* A, double\* B, double\* X, int N)

{

auto prevX = new double[N];

auto vectorR = new double[N];

auto tempVector = new double[N];

int itr = 1;

double normR = 0;

auto prevDelta = 1.0;

vectorClear(prevX, N);

printIterHeader();

do {

//R = Ax

matrixMulVec(A, prevX, vectorR, N);

//R = b - Ax

vectorSub(B, vectorR, vectorR, N);

//Ar

matrixMulVec(A, vectorR, tempVector, N);

//tau = <r, r> / <Ar, r>

double tau = vectorMul(vectorR, vectorR, N) / vectorMul(tempVector, vectorR, N);

//(b - Ax) \* tau

vectorMulScalar(vectorR, tau, tempVector, N);

//x(k+1) = x(k) + (b - Ax) \* tau

vectorAdd(prevX, tempVector, X, N);

//норма невязки

normR = calcVectorRNorm(A, B, X, prevX, tempVector, N);

//оценка нормы матрицы перехода q

auto q = calcQ(X, prevX, tempVector, &prevDelta, N);

//погрешность

auto err = calcError(X, prevX, tempVector, q, N);

//вывод

printIterStep(itr++, tau, q, normR, err, X, N);

copyVectorToVector(X, prevX, N);

} while (normR > eps);

}

//считаем alpha

double calcAlpha(double tau, double prevTau, double prevAlpha, double R2, double prevR2)

{

return 1.0 / (1.0 - R2 \* tau / (prevR2 \* prevTau \* prevAlpha));

}

//метод сопряженных градиентов

void ConjugateGradientMethod(double\*\* A, double\* B, double\* X, int N)

{

auto prevPrevX = new double[N];

auto prevX = new double[N];

auto vectorR = new double[N];

auto tempVector = new double[N];

int itr = 1;

double normR = 0;

auto prevAlpha = 1.0;

auto prevDelta = 1.0;

auto prevR2 = 0.0;

auto prevTau = 0.0;

vectorClear(prevX, N);

for (int i = 0; i < N; i++) prevPrevX[i] = 1.0;

printIterHeader();

do {

//R = Ax

matrixMulVec(A, prevX, vectorR, N);

//R = b - Ax

vectorSub(B, vectorR, vectorR, N);

//Ar

matrixMulVec(A, vectorR, tempVector, N);

//tau = <r, r> / <Ar, r>

double R2 = vectorMul(vectorR, vectorR, N);

double tau = R2 / vectorMul(tempVector, vectorR, N);

auto alpha = (itr == 1) ? 1.0 : calcAlpha(tau, prevTau, prevAlpha, R2, prevR2);

prevR2 = R2;

prevAlpha = alpha;

prevTau = tau;

//alpha \* x(k)

vectorMulScalar(prevX, alpha, X, N);

//(1 - alpha) \* x(k - 1)

vectorMulScalar(prevPrevX, 1 - alpha, tempVector, N);

vectorAdd(X, tempVector, X, N);

//alpha \* tau \* R

vectorMulScalar(vectorR, alpha \* tau, tempVector, N);

vectorAdd(X, tempVector, X, N);

//норма невязки

normR = calcVectorRNorm(A, B, X, prevX, tempVector, N);

//оценка нормы матрицы перехода q

auto q = calcQ(X, prevX, tempVector, &prevDelta, N);

//погрешность

auto err = calcError(X, prevX, tempVector, q, N);

//вывод

printIterStep(itr++, tau, q, normR, err, X, N, true, alpha);

copyVectorToVector(prevX, prevPrevX, N);

copyVectorToVector(X, prevX, N);

} while (normR > eps);

}

int SOR(double\*\* A, double\* B, double\* X, double omega, double N, double eps, bool printSteps);

double findBestOmega(double\*\* A, double\* B, int N) {

auto X = new double[N];

double omega = 0.1;

double bestOmega = 0.1;

int minItr = INT\_MAX;

while (omega < 2.0) {

auto itr = SOR(A, B, X, omega, N, 0.01, false);

if (itr < minItr)

{

minItr = itr;

bestOmega = omega;

}

printf("w = %3.1f\tItr = %d\n", omega, itr);

omega += 0.1;

}

printf("w\* = %3.1f\tItrMin = %d\n", bestOmega, minItr);

return bestOmega;

}

void SOR\_calcVectorX(double\*\* matrixA, double\* vectorXNext, double\* vectorX, double\* vectorB, int N, double omega)

{

for (int i = 0; i < N; i++)

{

double sumOne = 0;

double sumTwo = 0;

for (int j = 0; j < i; j++)

{

sumOne += matrixA[i][j] \* vectorXNext[j];

}

for (int j = i + 1; j < N; j++)

{

sumTwo += matrixA[i][j] \* vectorX[j];

}

double xWave = (vectorB[i] - sumOne - sumTwo) / matrixA[i][i];

vectorXNext[i] = vectorX[i] + omega \* (xWave - vectorX[i]);

}

}

//метод ПВР

int SOR(double\*\* A, double\* B, double\* X, double omega, double N, double eps, bool printSteps = true)

{

auto prevX = new double[N];

auto tempVector = new double[N];

int itr = 1;

double normR = 0;

auto prevDelta = 1.0;

vectorClear(prevX, N);

if (printSteps) {

printIterHeader();

}

do {

//расчитываем текущий вектор X

SOR\_calcVectorX(A, X, prevX, B, N, omega);

//норма невязки

normR = calcVectorRNorm(A, B, X, prevX, tempVector, N);

if (printSteps) {

//оценка нормы матрицы перехода q

auto q = calcQ(X, prevX, tempVector, &prevDelta, N);

//погрешность

auto err = calcError(X, prevX, tempVector, q, N);

//вывод

printIterStep(itr, omega, q, normR, err, X, N);

}

itr++;

copyVectorToVector(X, prevX, N);

} while (normR > eps);

return itr;

}

void initSomeStuff(double\*\* A, double\* vectorB, double\* vectorX, double\*\* backwardMatrix, int N)

{

//Матрица U

double\*\* U = nullptr;

createMatrix(&U, N);

copyMatrixToMatrix(A, U, N);

//Матрица L

double\*\* L = nullptr;

createMatrix(&L, N);

fillMatrixAsEmpty(L, N);

printf("1) Input\A:\n");

printMatrix(A, N);

printf("B:\n");

printVector(vectorB, N);

//LU разложение

int rank = N; //ранг матрицы

int\* P = new int[N]; //"матрица" перестановок (на самом деле подстановка)

double sign = 1.0;

LUdecomposition(L, U, P, rank, sign, N);

if (rank != N)

throw std::exception();

SolveSOLE(L, U, vectorX, P, vectorB, N);

SolveBackwardMatrix(L, U, backwardMatrix, P, N);

}

int main()

{

system("chcp 1251");

//чтение

double\*\* A = nullptr;

double\* B = nullptr;

int N;

readInputData("in2.txt", &A, &B, &N);

auto X\_LU = new double[N];

auto X\_Method1 = new double[N];

auto X\_Method2 = new double[N];

auto X\_Method3 = new double[N];

auto X\_Method4 = new double[N];

//обратная матрица

double\*\* backwardMatrix = nullptr;

createMatrix(&backwardMatrix, N);

//LU, ...

initSomeStuff(A, B, X\_LU, backwardMatrix, N);

//транспонируем матрицу

double\*\* trA = nullptr;

createMatrix(&trA, N);

copyMatrixToMatrix(A, trA, N);

transpose(trA, N);

double\*\* trBackA = nullptr;

createMatrix(&trBackA, N);

copyMatrixToMatrix(backwardMatrix, trBackA, N);

transpose(trBackA, N);

printf("Variant=\nb\n");

printVector(B, N);

printf("A\n");

printMatrix(A, N);

double euclidNorm = computEuclidNorm(A, trA, N);

double euclidNorm2 = computEuclidNorm(backwardMatrix, trBackA, N);

printf("Норма матрицы=%f\n", euclidNorm);

printf("Метод простой итерации\n");

SimpleIterationMethod(A, euclidNorm, B, X\_Method1, N);

printf("\nМетод наискорейшего спуска\n");

FastDescentMethod(A, B, X\_Method2, N);

printf("\nМетод ПВР - выбор оптимального w\n");

auto bestOmega = findBestOmega(A, B, N);

printf("\nМетод ПВР\n");

SOR(A, B, X\_Method3, bestOmega, N, eps);

printf("\nМетод сопряженных градиентов\n");

ConjugateGradientMethod(A, B, X\_Method4, N);

auto condA = euclidNorm \* euclidNorm2;

cout << "\nЧисло обусловленности " << condA << endl;

cout << "\nТеоретическая оценка чиcла итераций ";

cout << "\nМетод простых итераций " << (long)(log(1 / eps) / 2 \* condA);

cout << "\nМетод наискорейшего спуска " << (long)(log(1 / eps) \* condA);

cout << "\nМетод ПВР " << (long)(log(1 / eps) / 4 \* sqrt(condA));

cout << "\nМетод сопряженных градиентов " << (long)(log(2 / eps) / 2 \* sqrt(condA));

auto tempVector = new double[N];

cout << "\n\nСравнение с LU разложением\n";

printVector(X\_LU, 0);

cout << "\nРазница между LU разложением и методом простых итераций\n";

vectorSub(X\_LU, X\_Method1, tempVector, N);

printVector(tempVector, N, true);

cout << "\nРазница между LU разложением и методом наискорейшего спуска\n";

vectorSub(X\_LU, X\_Method2, tempVector, N);

printVector(tempVector, N, true);

cout << "\nРазница между LU разложением и методом ПВР\n";

vectorSub(X\_LU, X\_Method3, tempVector, N);

printVector(tempVector, N, true);

cout << "\nРазница между LU разложением и методом сопряженных градиентов\n";

vectorSub(X\_LU, X\_Method4, tempVector, N);

printVector(tempVector, N, true);

return 0;

}