|  |  |
| --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ПЕРМСКОГО КРАЯ | |
| Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  «Пермский государственный национальный исследовательский университет» | |
| **ЧИСЛЕННЫЕ МЕТОДЫ  «ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАЧЕСКИХ УРАВНЕНИЙ»** | |
| *Лабораторная работа №3* | |
|  | |
|  | Работу выполнили студенты группы ПМИ-1-2015 3 курса механико-математического факультета  Толов Я.Ф., Заманов М.Р. |
|  | Проверил:  Профессор, доктор физико-математических наук  С. В. Русаков  “\_\_\_\_” 20\_\_ г. |
|  |  |
| Пермь 2017 | |

*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.

3) Оценку нормы матрицы перехода осуществлять по формуле



4) 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.

*Исходные данные*

Варианты исходных данных: 14 a) и 20 a).

Дана матрица А.

Вариант 14:

a)┌ ┐ b = [1,2,3,4]

│ 8.2 -3.3 1.0 -2.4│

│-3.3 6.4 -4.0 0.1│

│ 1.0 -4.0 6.9 1.4│

│-2.4 0.1 1.4 1.2│

└ ┘

Вариант 20:

a)┌ ┐ b = [1,2,3,4]

│ 4.3 1.9 -2.9 -1.2│

│ 1.9 4.4 2.0 4.3│

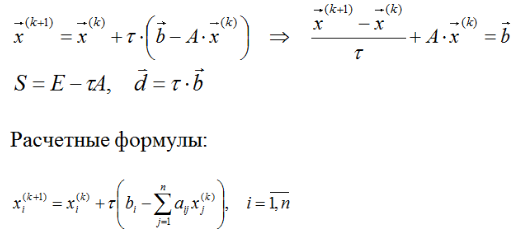
│-2.9 2.0 9.8 2.4│

│-1.2 4.3 2.4 9.3│

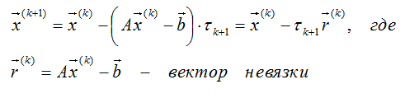
└ ┘

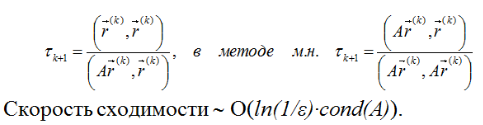
***Решение***

1. Решение методом простой итерации:

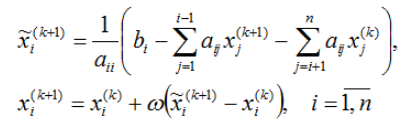


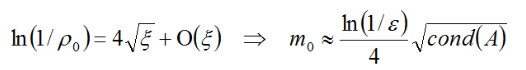
1. Методом наискорейшего градиентного спуска:



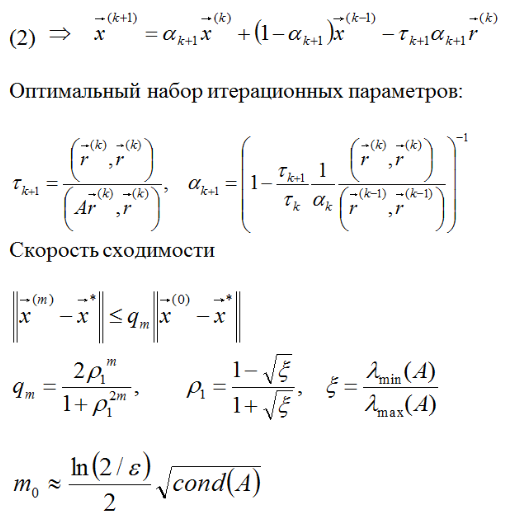


1. Методом ПВР:

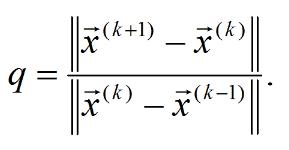




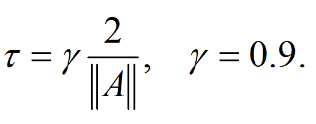
1. Метод сопряженных градиентов:



Оценка нормы матрицы перехода:



Значение итерационного параметра в методе простой итерации:



***Краткие выводы***

Решая СЛАУ 4 разными итерационными методами, мы пришли к следующим выводам:

1. Разные методы решения несут за собой разное количество экономических, временных, трудовых затрат.

2. Так как все методы дают ответ с заданной точностью, рядовому пользователю может быть без разницы то, каким методом СЛАУ решать.

3. Для работы с такими массивами данных требуется большой объём памяти.

4. Для решения СЛАУ 4-мя представленными методами требуются знания в области высшей линейной алгебры.

Решение, полученное итерационным методом, отличается от решения прямым методом тем, что прямое решение имеет определённого количество шагов выполнения программы и вычисляет более точно, в то время как итерационный метод имеет неопределённое количество шагов и получает решение с определённой точностью.

***Анализ эффективности выбранных методов***

Метод простых итерации является самым медленным методом среди представленных, также является одним из самых неточных методов.

Градиентный метод наискорейшего спуска имеет более высокую скорость сходимости по сравнению с методом простых итераций.

У ПВР-метода меньше итераций, чем у градиентного метода или метода простых итераций, но для его реализации требуются дополнительных действий для нахождения оптимального омега.

Метод сопряженных градиентов сходится быстрее всех остальных методов и является самым точным, в то же время этот метод является наиболее трудоёмким в плане вычислительных затрат.

***Сравнение фактического числа итерации с теоретическим***

Для подсчёта теоретического числа итераций использовались следующие формулы:

Метод простых итераций: m0 = (log(1/ε)\*cond(A))/2

Градиентный метод наискорейшего спуска: m0 = log(1/ε)\*cond(A)

ПВР метод: m0 = (log(1/ε)\*sqrt(cond(A)))/4

Метод сопряженных градиентов: m0 = (log(2/ε)\*sqrt(cond(A)))/2

Вариант 20

Cond(A) = 246.9

Метод: Фактическое число итераций Теоретическое число итераций

Простых ит. 663 1137

Градиентный 416 2274

ПВР 47 36.18

Сопряж. Град. 5 77.78

Вариант 14

Cond(A) = 316.98

Метод: Фактическое число итераций Теоретическое число итераций

Простых ит. 730 1459.69

Градиентный 561 2919.38

ПВР 27 41

Сопряж. Град. 5 88.13

***Результаты тестирования***

Входные данные: Вариант 20

a)┌ ┐ b = [1,2,3,4]

│ 4.3 1.9 -2.9 -1.2│

│ 1.9 4.4 2.0 4.3│

│-2.9 2.0 9.8 2.4│

│-1.2 4.3 2.4 9.3│

└ ┘

Метод простой итерации:

0 X = 1.66977 -1.33837 -0.88256 -1.00233 norm r = 5.00233 |A\*x-b| = 23.19849

1 X = 0.89542 -0.20904 1.37516 1.42542 norm r = 2.42775 |A\*x-b| = 10.58343

q = 1.01459 2 X = 1.23505 -1.01079 0.23626 0.31785 norm r = 1.13889 |A\*x-b| = 6.30537

q = 1.05401 3 X = 1.09653 -0.77411 0.81446 0.97772 norm r = 0.65986 |A\*x-b| = 2.40297

q = 1.06182 4 X = 1.23163 -1.03683 0.54236 0.72625 norm r = 0.27210 |A\*x-b| = 1.88058

q = 1.06262 5 X = 1.24401 -1.03533 0.70847 0.92305 norm r = 0.19681 |A\*x-b| = 0.33999

q = 1.06059 6 X = 1.32565 -1.16030 0.65823 0.88747 norm r = 0.12498 |A\*x-b| = 0.79945

q = 1.05772 7 X = 1.37568 -1.21744 0.71939 0.96563 norm r = 0.07816 |A\*x-b| = 0.38965

q = 1.05457 8 X = 1.44294 -1.30619 0.72534 0.98436 norm r = 0.08875 |A\*x-b| = 0.58534

q = 1.05153 9 X = 1.50174 -1.37712 0.75947 1.03174 norm r = 0.07093 |A\*x-b| = 0.47738

q = 1.04866 10 X = 1.56449 -1.45555 0.77938 1.06374 norm r = 0.07843 |A\*x-b| = 0.52356

q = 1.04602 11 X = 1.62466 -1.52890 0.80630 1.10277 norm r = 0.07336 |A\*x-b| = 0.49252

q = 1.04361 12 X = 1.68540 -1.60364 0.82942 1.13762 norm r = 0.07474 |A\*x-b| = 0.50074

q = 1.04141 13 X = 1.74507 -1.67657 0.85415 1.17400 norm r = 0.07293 |A\*x-b| = 0.48934

q = 1.03939 14 X = 1.80446 -1.74933 0.87775 1.20907 norm r = 0.07276 |A\*x-b| = 0.48796

q = 15-652 X = norm r = |A\*x-b| =

q = 1.00002 653 X = 7.96678 -9.28703 3.36277 4.88419 norm r = 0.00016 |A\*x-b| = 0.00108

q = 1.00002 654 X = 7.96691 -9.28719 3.36282 4.88427 norm r = 0.00016 |A\*x-b| = 0.00107

q = 1.00002 655 X = 7.96704 -9.28734 3.36288 4.88434 norm r = 0.00016 |A\*x-b| = 0.00106

q = 1.00002 656 X = 7.96717 -9.28750 3.36293 4.88442 norm r = 0.00016 |A\*x-b| = 0.00105

q = 1.00002 657 X = 7.96730 -9.28766 3.36298 4.88450 norm r = 0.00016 |A\*x-b| = 0.00104

q = 1.00002 658 X = 7.96742 -9.28781 3.36303 4.88457 norm r = 0.00015 |A\*x-b| = 0.00103

q = 1.00002 659 X = 7.96755 -9.28796 3.36308 4.88465 norm r = 0.00015 |A\*x-b| = 0.00102

q = 1.00002 660 X = 7.96767 -9.28811 3.36313 4.88472 norm r = 0.00015 |A\*x-b| = 0.00101

q = 1.00002 661 X = 7.96779 -9.28826 3.36318 4.88479 norm r = 0.00015 |A\*x-b| = 0.00100

q = 1.00002 662 X = 7.96792 -9.28841 3.36323 4.88486 norm r = 0.00015 |A\*x-b| = 0.00099

Ответ: VIM X = 7.96792 -9.28841 3.36323 4.88486

Градиентный метод наискорейшего спуска:

0 X = 1.45027 -0.24430 0.38986 0.63707 tau = 0.07035 norm r = 3.36293 |A\*x-b| = 22.88060

1 X = 1.06492 -0.67289 0.70393 0.62774 tau = 0.13395 norm r = 0.42859 |A\*x-b| = 1.21989

q = 0.92720 2 X = 1.16238 -0.90366 0.51236 0.75477 tau = 0.19728 norm r = 0.23077 |A\*x-b| = 1.65049

q = 1.07924 3 X = 1.17522 -0.96220 0.66907 0.87491 tau = 0.11646 norm r = 0.15671 |A\*x-b| = 0.26948

q = 1.09408 4 X = 1.30535 -1.14924 0.61393 0.84179 tau = 0.17012 norm r = 0.18704 |A\*x-b| = 1.20609

q = 1.06956 5 X = 1.34739 -1.18080 0.73585 0.98221 tau = 0.11643 norm r = 0.14042 |A\*x-b| = 0.21208

q = 1.08943 6 X = 1.47709 -1.36109 0.68488 0.94711 tau = 0.17013 norm r = 0.18029 |A\*x-b| = 1.17335

q = 1.05906 7 X = 1.51869 -1.39136 0.80374 1.08372 tau = 0.11643 norm r = 0.13661 |A\*x-b| = 0.20584

q = 1.07565 8 X = 1.64536 -1.56677 0.75429 1.04931 tau = 0.17013 norm r = 0.17540 |A\*x-b| = 1.14270

q = 1.05060 9 X = 1.68592 -1.59622 0.87010 1.18235 tau = 0.11643 norm r = 0.13304 |A\*x-b| = 0.20047

q = 1.06522 10 X = 1.80935 -1.76707 0.82196 1.14880 tau = 0.17013 norm r = 0.17084 |A\*x-b| = 1.11309

q = 1.04405 11 X = 1.84887 -1.79575 0.93477 1.27839 tau = 0.11643 norm r = 0.12959 |A\*x-b| = 0.19527

q = 1.05712 12 X = 1.96910 -1.96217 0.88787 1.24571 tau = 0.17013 norm r = 0.16642 |A\*x-b| = 1.08427

q = 1.03888 13 X = 2.00760 -1.99012 0.99777 1.37195 tau = 0.11643 norm r = 0.12624 |A\*x-b| = 0.19022

q = 1.05067 14 X = 2.12472 -2.15223 0.95208 1.34011 tau = 0.17013 norm r = 0.16211 |A\*x-b| = 1.05620

q = 15-405 X = tau = norm r = |A\*x-b| =

q = 1.00007 406 X = 7.94624 -9.26197 3.35417 4.87159 tau = 0.17013 norm r = 0.00095 |A\*x-b| = 0.00618

q = 1.00005 407 X = 7.94646 -9.26213 3.35479 4.87231 tau = 0.11643 norm r = 0.00072 |A\*x-b| = 0.00108

q = 1.00007 408 X = 7.94713 -9.26305 3.35453 4.87213 tau = 0.17013 norm r = 0.00092 |A\*x-b| = 0.00602

q = 1.00005 409 X = 7.94734 -9.26321 3.35514 4.87283 tau = 0.11643 norm r = 0.00070 |A\*x-b| = 0.00106

q = 1.00006 410 X = 7.94799 -9.26411 3.35489 4.87265 tau = 0.17013 norm r = 0.00090 |A\*x-b| = 0.00587

q = 1.00005 411 X = 7.94820 -9.26426 3.35549 4.87333 tau = 0.11643 norm r = 0.00068 |A\*x-b| = 0.00103

q = 1.00006 412 X = 7.94884 -9.26514 3.35524 4.87316 tau = 0.17013 norm r = 0.00088 |A\*x-b| = 0.00571

q = 1.00005 413 X = 7.94904 -9.26528 3.35582 4.87383 tau = 0.11643 norm r = 0.00067 |A\*x-b| = 0.00100

q = 1.00006 414 X = 7.94966 -9.26614 3.35558 4.87366 tau = 0.17013 norm r = 0.00085 |A\*x-b| = 0.00557

q = 1.00004 415 X = 7.94985 -9.26628 3.35614 4.87431 tau = 0.11643 norm r = 0.00065 |A\*x-b| = 0.00098

Ответ: SDM X = 7.94985 -9.26628 3.35614 4.87431

Метод ПВР: Используется omega = 1.61

0 X = 3.39628 -11.33846 2.42929 6.38914 norm r = 13.33846 |A\*x-b| = 22.23629

1 X = 11.87723 -12.43962 6.23780 5.93095 norm r = 8.48095 |A\*x-b| = 14.14115

q = 1.60339 2 X = 11.41668 -13.51395 4.22884 7.74920 norm r = 2.00896 |A\*x-b| = 7.45935

q = 1.00208 3 X = 11.09747 -14.02734 4.75400 6.73773 norm r = 1.01147 |A\*x-b| = 3.56401

q = 0.94617 4 X = 11.77319 -12.97682 4.80922 6.69015 norm r = 1.05052 |A\*x-b| = 1.75130

q = 1.02088 5 X = 10.65224 -12.80386 4.20342 6.60925 norm r = 1.12095 |A\*x-b| = 2.24937

q = 0.93169 6 X = 10.51884 -12.24601 4.35800 6.15139 norm r = 0.55785 |A\*x-b| = 1.61333

q = 0.97179 7 X = 10.16549 -11.73336 4.10745 6.07976 norm r = 0.51265 |A\*x-b| = 0.93032

q = 0.97070 8 X = 9.71210 -11.43480 3.97442 5.86229 norm r = 0.45339 |A\*x-b| = 0.76628

q = 0.96137 9 X = 9.53413 -11.05367 3.93129 5.69217 norm r = 0.38113 |A\*x-b| = 0.63538

q = 0.97865 10 X = 9.24829 -10.78822 3.80127 5.59298 norm r = 0.28584 |A\*x-b| = 0.48277

q = 0.97469 11 X = 9.04806 -10.55971 3.74922 5.46342 norm r = 0.22850 |A\*x-b| = 0.45652

q = 0.97927 12 X = 8.89292 -10.34930 3.68901 5.37860 norm r = 0.21042 |A\*x-b| = 0.35078

q = 0.98378 13 X = 8.73438 -10.18991 3.63128 5.30274 norm r = 0.15939 |A\*x-b| = 0.26731

q = 0.98418 14 X = 8.62093 -10.04666 3.59529 5.23377 norm r = 0.14325 |A\*x-b| = 0.24305

q = 0.98763 15 X = 8.51815 -9.92772 3.55639 5.18211 norm r = 0.11894 |A\*x-b| = 0.19828

q = 0.98918 16 X = 8.43079 -9.82980 3.52669 5.13492 norm r = 0.09792 |A\*x-b| = 0.16628

q = 0.99062 17 X = 8.36097 -9.74500 3.50229 5.09622 norm r = 0.08480 |A\*x-b| = 0.14136

q = 0.99231 18 X = 8.29935 -9.67513 3.48012 5.06422 norm r = 0.06987 |A\*x-b| = 0.11648

q = 0.99331 19 X = 8.24879 -9.61604 3.46275 5.03646 norm r = 0.05909 |A\*x-b| = 0.09851

q = 0.99439 20 X = 8.20626 -9.56613 3.44763 5.01369 norm r = 0.04991 |A\*x-b| = 0.08320

q = 0.99528 21 X = 8.17005 -9.52451 3.43490 4.99436 norm r = 0.04162 |A\*x-b| = 0.06939

q = 0.99597 22 X = 8.14002 -9.48930 3.42441 4.97806 norm r = 0.03521 |A\*x-b| = 0.05870

q = 0.99663 23 X = 8.11458 -9.45976 3.41541 4.96447 norm r = 0.02954 |A\*x-b| = 0.04924

q = 0.99714 24 X = 8.09321 -9.43495 3.40793 4.95296 norm r = 0.02481 |A\*x-b| = 0.04136

q = 0.99759 25 X = 8.07530 -9.41405 3.40163 4.94332 norm r = 0.02090 |A\*x-b| = 0.03485

q = 0.99797 26 X = 8.06018 -9.39650 3.39631 4.93521 norm r = 0.01754 |A\*x-b| = 0.02925

q = 0.99829 27 X = 8.04750 -9.38174 3.39186 4.92838 norm r = 0.01476 |A\*x-b| = 0.02461

q = 0.99856 28 X = 8.03683 -9.36933 3.38810 4.92265 norm r = 0.01241 |A\*x-b| = 0.02069

q = 0.99879 29 X = 8.02786 -9.35890 3.38495 4.91783 norm r = 0.01043 |A\*x-b| = 0.01739

q = 0.99898 30 X = 8.02032 -9.35013 3.38230 4.91378 norm r = 0.00877 |A\*x-b| = 0.01463

q = 0.99914 31 X = 8.01398 -9.34276 3.38007 4.91037 norm r = 0.00737 |A\*x-b| = 0.01229

q = 0.99928 32 X = 8.00865 -9.33656 3.37820 4.90751 norm r = 0.00620 |A\*x-b| = 0.01033

q = 0.99939 33 X = 8.00417 -9.33134 3.37663 4.90510 norm r = 0.00521 |A\*x-b| = 0.00869

q = 0.99949 34 X = 8.00041 -9.32696 3.37530 4.90307 norm r = 0.00438 |A\*x-b| = 0.00730

q = 0.99957 35 X = 7.99724 -9.32328 3.37419 4.90137 norm r = 0.00368 |A\*x-b| = 0.00614

q = 0.99964 36 X = 7.99458 -9.32018 3.37325 4.89994 norm r = 0.00310 |A\*x-b| = 0.00516

q = 0.99970 37 X = 7.99234 -9.31758 3.37247 4.89873 norm r = 0.00260 |A\*x-b| = 0.00434

q = 0.99974 38 X = 7.99046 -9.31539 3.37180 4.89772 norm r = 0.00219 |A\*x-b| = 0.00365

q = 0.99978 39 X = 7.98887 -9.31355 3.37125 4.89687 norm r = 0.00184 |A\*x-b| = 0.00307

q = 0.99982 40 X = 7.98754 -9.31200 3.37078 4.89616 norm r = 0.00155 |A\*x-b| = 0.00258

q = 0.99985 41 X = 7.98642 -9.31070 3.37039 4.89556 norm r = 0.00130 |A\*x-b| = 0.00217

q = 0.99987 42 X = 7.98548 -9.30960 3.37006 4.89505 norm r = 0.00109 |A\*x-b| = 0.00182

q = 0.99989 43 X = 7.98469 -9.30868 3.36978 4.89462 norm r = 0.00092 |A\*x-b| = 0.00153

q = 0.99991 44 X = 7.98403 -9.30791 3.36954 4.89427 norm r = 0.00077 |A\*x-b| = 0.00129

q = 0.99992 45 X = 7.98347 -9.30726 3.36935 4.89397 norm r = 0.00065 |A\*x-b| = 0.00108

q = 0.99994 46 X = 7.98300 -9.30671 3.36918 4.89371 norm r = 0.00055 |A\*x-b| = 0.00091

Ответ: SOR X = 7.98300 -9.30671 3.36918 4.89371

Метод сопряжённых градиентов:

0 X = 1.45027 -0.24430 0.38986 0.63707 norm r = 3.36293 |A\*x-b| = 22.88060

1 X = 1.06555 -0.69904 0.68147 0.59475 norm r = 0.45474 |A\*x-b| = 1.22263

q = 1.03510 2 X = 1.11461 -1.15920 0.53692 1.00043 norm r = 0.46016 |A\*x-b| = 1.86150

q = 4.68296 3 X = 7.98052 -9.30383 3.36831 4.89238 norm r = 8.14463 |A\*x-b| = 28.35615

q = 1.65902 4 X = 7.98052 -9.30383 3.36831 4.89238 norm r = 0.00000 |A\*x-b| = 0.00000

Ответ: CGM X = 7.98052 -9.30383 3.36831 4.89238

Входные данные: Вариант 14

a)┌ ┐ b = [1,2,3,4]

│ 8.2 -3.3 1.0 -2.4│

│-3.3 6.4 -4.0 0.1│

│ 1.0 -4.0 6.9 1.4│

│-2.4 0.1 1.4 1.2│

└ ┘

Метод простой итерации:

0 X = 1.72483 2.49530 1.03087 3.66174 norm r = 1.96913 |A\*x-b| = 3.32201

1 X = 2.06891 1.94918 0.91211 3.90973 norm r = 0.54612 |A\*x-b| = 2.80183

q = 0.99744 2 X = 1.94067 1.90208 0.54494 4.24820 norm r = 0.36717 |A\*x-b| = 2.60664

q = 1.20334 3 X = 2.06318 1.65876 0.41931 4.56310 norm r = 0.31490 |A\*x-b| = 2.72300

q = 1.07872 4 X = 2.07381 1.58789 0.21277 4.89205 norm r = 0.32895 |A\*x-b| = 2.65001

q = 1.10941 5 X = 2.16598 1.47227 0.08722 5.21219 norm r = 0.32014 |A\*x-b| = 2.67438

q = 1.07357 6 X = 2.22874 1.41826 -0.05482 5.53527 norm r = 0.32308 |A\*x-b| = 2.64157

q = 1.07609 7 X = 2.31862 1.35848 -0.16679 5.85439 norm r = 0.31912 |A\*x-b| = 2.63709

q = 1.06307 8 X = 2.40169 1.32279 -0.27914 6.17296 norm r = 0.31857 |A\*x-b| = 2.61501

q = 1.06041 9 X = 2.49418 1.28967 -0.37900 6.48887 norm r = 0.31591 |A\*x-b| = 2.60102

q = 1.05425 10 X = 2.58550 1.26696 -0.47622 6.80308 norm r = 0.31422 |A\*x-b| = 2.58151

q = 1.05118 11 X = 2.68015 1.24744 -0.56755 7.11494 norm r = 0.31186 |A\*x-b| = 2.56426

q = 1.04749 12 X = 2.77471 1.23284 -0.65637 7.42472 norm r = 0.30978 |A\*x-b| = 2.54527

q = 1.04488 13 X = 2.87032 1.22057 -0.74202 7.73220 norm r = 0.30748 |A\*x-b| = 2.52690

q = 1.04229 14 X = 2.96582 1.21080 -0.82576 8.03747 norm r = 0.30526 |A\*x-b| = 2.50800

q = 15-719 X = norm r = |A\*x-b| =

q = 1.00003 720 X = 15.59455 0.75624 -10.97422 47.25374 norm r = 0.00130 |A\*x-b| = 0.01071

q = 1.00003 721 X = 15.59497 0.75622 -10.97455 47.25504 norm r = 0.00129 |A\*x-b| = 0.01063

q = 1.00003 722 X = 15.59538 0.75621 -10.97488 47.25632 norm r = 0.00128 |A\*x-b| = 0.01055

q = 1.00003 723 X = 15.59579 0.75619 -10.97521 47.25759 norm r = 0.00127 |A\*x-b| = 0.01046

q = 1.00003 724 X = 15.59620 0.75618 -10.97554 47.25886 norm r = 0.00126 |A\*x-b| = 0.01038

q = 1.00003 725 X = 15.59660 0.75616 -10.97586 47.26011 norm r = 0.00125 |A\*x-b| = 0.01030

q = 1.00003 726 X = 15.59701 0.75615 -10.97618 47.26136 norm r = 0.00124 |A\*x-b| = 0.01022

q = 1.00003 727 X = 15.59740 0.75614 -10.97650 47.26259 norm r = 0.00124 |A\*x-b| = 0.01015

q = 1.00003 728 X = 15.59780 0.75612 -10.97682 47.26382 norm r = 0.00123 |A\*x-b| = 0.01007

q = 1.00003 729 X = 15.59819 0.75611 -10.97713 47.26504 norm r = 0.00122 |A\*x-b| = 0.00999

Ответ: VIM X = 15.59819 0.75611 -10.97713 47.26504

Градиентный метод наискорейшего спуска:

0 X = 1.74499 2.50908 0.97610 3.65234 tau = 0.12417 norm r = 2.02390 |A\*x-b| = 3.30037

1 X = 2.14630 1.81704 0.89512 3.97037 tau = 0.14537 norm r = 0.69204 |A\*x-b| = 3.16774

q = 1.01794 2 X = 1.85188 1.91228 0.35506 4.41160 tau = 0.14948 norm r = 0.54006 |A\*x-b| = 2.46229

q = 1.29460 3 X = 2.20900 1.43544 0.38097 4.78453 tau = 0.15146 norm r = 0.47684 |A\*x-b| = 2.88326

q = 1.02960 4 X = 2.01270 1.61225 -0.04926 5.22847 tau = 0.15397 norm r = 0.44393 |A\*x-b| = 2.46406

q = 1.19080 5 X = 2.38476 1.24288 0.02108 5.60826 tau = 0.15414 norm r = 0.37980 |A\*x-b| = 2.83971

q = 1.01927 6 X = 2.22704 1.46649 -0.35352 6.04961 tau = 0.15542 norm r = 0.44135 |A\*x-b| = 2.43365

q = 1.14587 7 X = 2.60648 1.14821 -0.25925 6.42648 tau = 0.15486 norm r = 0.37944 |A\*x-b| = 2.79191

q = 1.01321 8 X = 2.46609 1.39328 -0.60537 6.86139 tau = 0.15577 norm r = 0.43491 |A\*x-b| = 2.39313

q = 1.12019 9 X = 2.84568 1.10082 -0.50024 7.23238 tau = 0.15502 norm r = 0.37959 |A\*x-b| = 2.74103

q = 1.00999 10 X = 2.71416 1.35350 -0.83004 7.65958 tau = 0.15585 norm r = 0.42720 |A\*x-b| = 2.34917

q = 1.10299 11 X = 3.08995 1.07566 -0.72092 8.02385 tau = 0.15506 norm r = 0.37580 |A\*x-b| = 2.68899

q = 1.00816 12 X = 2.96373 1.32919 -1.03989 8.44299 tau = 0.15587 norm r = 0.41914 |A\*x-b| = 2.30430

q = 1.09033 13 X = 3.33385 1.06086 -0.93009 8.80032 tau = 0.15507 norm r = 0.37012 |A\*x-b| = 2.63690

q = 1.00701 14 X = 3.21135 1.31211 -1.24079 9.21135 tau = 0.15588 norm r = 0.41103 |A\*x-b| = 2.25951

q = 15-550 X = tau = norm r = |A\*x-b| =

q = 1.00007 551 X = 15.59067 0.75578 -10.97014 47.23982 tau = 0.15507 norm r = 0.00175 |A\*x-b| = 0.01243

q = 1.00001 552 X = 15.59009 0.75697 -10.97159 47.24176 tau = 0.15588 norm r = 0.00194 |A\*x-b| = 0.01065

q = 1.00007 553 X = 15.59181 0.75575 -10.97108 47.24341 tau = 0.15507 norm r = 0.00172 |A\*x-b| = 0.01218

q = 1.00001 554 X = 15.59125 0.75692 -10.97250 47.24531 tau = 0.15588 norm r = 0.00190 |A\*x-b| = 0.01044

q = 1.00007 555 X = 15.59293 0.75572 -10.97199 47.24693 tau = 0.15507 norm r = 0.00168 |A\*x-b| = 0.01194

q = 1.00001 556 X = 15.59238 0.75687 -10.97339 47.24879 tau = 0.15588 norm r = 0.00186 |A\*x-b| = 0.01023

q = 1.00007 557 X = 15.59403 0.75570 -10.97289 47.25038 tau = 0.15507 norm r = 0.00165 |A\*x-b| = 0.01171

q = 1.00001 558 X = 15.59349 0.75682 -10.97427 47.25220 tau = 0.15588 norm r = 0.00182 |A\*x-b| = 0.01003

q = 1.00007 559 X = 15.59511 0.75567 -10.97378 47.25376 tau = 0.15507 norm r = 0.00162 |A\*x-b| = 0.01148

q = 1.00001 560 X = 15.59458 0.75678 -10.97512 47.25554 tau = 0.15588 norm r = 0.00179 |A\*x-b| = 0.00983

Ответ: SDM X = 15.59458 0.75678 -10.97512 47.25554

Метод ПВР: Используется omega = 1.56

0 X = 2.14146 3.91754 0.79084 7.69277 norm r = 3.69277 |A\*x-b| = 5.47193

1 X = 4.81243 2.74823 -0.80221 17.00957 norm r = 9.31681 |A\*x-b| = 7.86222

q = 1.18481 2 X = 7.13957 3.49463 -2.71021 22.42839 norm r = 5.41881 |A\*x-b| = 6.85016

q = 1.24450 3 X = 9.14210 2.69504 -4.53275 29.06270 norm r = 6.63432 |A\*x-b| = 5.89462

q = 1.11767 4 X = 10.89456 2.61376 -6.08175 33.64491 norm r = 4.58221 |A\*x-b| = 5.15853

q = 1.09756 5 X = 12.24901 2.12679 -7.41129 37.78782 norm r = 4.14290 |A\*x-b| = 3.98694

q = 1.06047 6 X = 13.32933 1.87119 -8.45346 40.76837 norm r = 2.98055 |A\*x-b| = 3.18001

q = 1.04397 7 X = 14.12302 1.56398 -9.27052 43.10258 norm r = 2.33421 |A\*x-b| = 2.33632

q = 1.02852 8 X = 14.70689 1.35214 -9.87538 44.74547 norm r = 1.64290 |A\*x-b| = 1.71867

q = 1.01914 9 X = 15.11213 1.16695 -10.31577 45.91536 norm r = 1.16989 |A\*x-b| = 1.19284

q = 1.01190 10 X = 15.38686 1.03376 -10.62201 46.69209 norm r = 0.77673 |A\*x-b| = 0.80872

q = 1.00716 11 X = 15.56230 0.93194 -10.82811 47.19280 norm r = 0.50071 |A\*x-b| = 0.51640

q = 1.00384 12 X = 15.66796 0.86080 -10.95941 47.49027 norm r = 0.29747 |A\*x-b| = 0.32521

q = 1.00171 13 X = 15.72492 0.81119 -11.03777 47.65050 norm r = 0.16023 |A\*x-b| = 0.19411

q = 1.00036 14 X = 15.74995 0.77879 -11.07957 47.71914 norm r = 0.06863 |A\*x-b| = 0.10352

q = 0.99959 15 X = 15.75488 0.75847 -11.09737 47.73112 norm r = 0.02031 |A\*x-b| = 0.04666

q = 0.99920 16 X = 15.74822 0.74685 -11.10020 47.71031 norm r = 0.02080 |A\*x-b| = 0.02671

q = 0.99905 17 X = 15.73569 0.74102 -11.09447 47.67318 norm r = 0.03713 |A\*x-b| = 0.03689

q = 0.99905 18 X = 15.72101 0.73897 -11.08446 47.63021 norm r = 0.04297 |A\*x-b| = 0.04323

q = 0.99913 19 X = 15.70642 0.73919 -11.07297 47.58782 norm r = 0.04239 |A\*x-b| = 0.04294

q = 0.99926 20 X = 15.69318 0.74066 -11.06167 47.54950 norm r = 0.03832 |A\*x-b| = 0.03896

q = 0.99939 21 X = 15.68187 0.74269 -11.05147 47.51684 norm r = 0.03265 |A\*x-b| = 0.03330

q = 0.99952 22 X = 15.67263 0.74486 -11.04280 47.49024 norm r = 0.02661 |A\*x-b| = 0.02719

q = 0.99963 23 X = 15.66537 0.74691 -11.03574 47.46936 norm r = 0.02088 |A\*x-b| = 0.02138

q = 0.99973 24 X = 15.65985 0.74871 -11.03020 47.45351 norm r = 0.01584 |A\*x-b| = 0.01625

q = 0.99981 25 X = 15.65578 0.75022 -11.02601 47.44188 norm r = 0.01163 |A\*x-b| = 0.01196

q = 0.99987 26 X = 15.65289 0.75142 -11.02293 47.43362 norm r = 0.00826 |A\*x-b| = 0.00851

Ответ: SOR X = 15.65289 0.75142 -11.02293 47.43362

Метод сопряжённых градиентов:

0 X = 1.74499 2.50908 0.97610 3.65234 norm r = 2.02390 |A\*x-b| = 3.30037

1 X = 2.31294 1.79044 0.58911 3.96606 norm r = 0.71864 |A\*x-b| = 3.78799

q = 2.12081 2 X = 2.02240 -0.07300 -1.48809 7.02342 norm r = 3.05735 |A\*x-b| = 11.10097

q = 9.63797 3 X = 15.64870 0.75437 -11.01764 47.42178 norm r = 40.39836 |A\*x-b| = 109.21593

q = 0.69986 4 X = 15.64870 0.75437 -11.01764 47.42178 norm r = 0.00000 |A\*x-b| = 0.00000

Ответ: CGM X = 15.64870 0.75437 -11.01764 47.42178

***Код на C++***

#include <iostream>

#include <iomanip>

#include "Matrix.h"

using namespace std;

double\*\* init(int n, int m, bool fillByZero);

double\* ValueIterationMethod(Matrix C, double\* X);

void copy(double\* A, double\* B, int n);

double Max(double\* E, int n);

double Norm(double\*\* M, int n);

double scalar(double\* A, double\* B, int n);

double GetQ(double\* X\_1, double\* X0, double\* X, int n);

double\* VIM(Matrix C, double\* X); // - методом простой итерации

double\* SOR(Matrix C, double\* X); //- методом ПВР

double\* CGM(Matrix C, double\* X); //- методом сопряженных градиентов

double\* SDM(Matrix C, double \* X); //- градиентным методом наискорейшего спуска

const double c\_delta = 0.001;

double teta;

const bool OUT\_VIM = true;

const bool OUT\_SOR = true;

const bool OUT\_CGM = true;

const bool OUT\_SDM = true;

double w = 1.61;

class Matrix

{

private:

public:

double\*\* value;

double\* b;

int n;

Matrix(int n, bool fillByZero = false)

{

this->n = n;

value = new double\*[n];

for (int i = 0; i < n; i++)

value[i] = new double[n];

b = new double[n];

if (fillByZero)

{

for (int i = 0; i < n; i++, b[i] = 0)

for (int j = 0; j < n; j++)

value[i][j] = 0;

}

}

Matrix(const char\* fileName)

{

std::ifstream fin(fileName);

fin >> n;

value = new double\*[n];

for (int i = 0; i < n; i++)

value[i] = new double[n];

b = new double[n];

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

fin >> value[i][j];

for (int i = 0; i < n; i++)

fin >> b[i];

fin.close();

}

void Print()

{

using std::cout;

using std::endl;

using std::setprecision;

using std::ios;

using std::setw;

using std::fixed;

cout.setf(ios::left);

for (int i = 0; i < n; i++, cout << endl)

for (int j = 0; j < n; j++)

{

if (value[i][j] > 0)

cout << " ";

cout << fixed << setprecision(1) << value[i][j] << " ";

}

cout.unsetf(ios::left);

}

void Delete()

{

for (int i = 0; i < n; i++)

delete[] value[i];

delete[] value;

delete[] b;

}

};

int main()

{

Matrix A("input20.txt");

teta = 0.9 \* 2 / Norm(A.value, A.n);

cout << "A:\n";

A.Print();

cout << "b = ";

for (int i = 0; i < A.n; i++)

cout << A.b[i] << " ";

cout << endl;

double\* X = new double[A.n];

for (int i = 0; i < A.n; i++)

X[i] = A.b[i];

X = VIM(A, X);

cout << "VIM X = ";

for (int i = 0; i < A.n; i++)

cout << fixed << setprecision(5) << X[i] << " ";

cout << endl;

for (int i = 0; i < A.n; i++)

X[i] = A.b[i];

X = SDM(A, X);

cout << "SDM X = ";

for (int i = 0; i < A.n; i++)

cout << fixed << setprecision(5) << X[i] << " ";

cout << endl;

for (int i = 0; i < A.n; i++)

X[i] = A.b[i];

X = SOR(A, X);

cout << "SOR X = ";

for (int i = 0; i < A.n; i++)

cout << fixed << setprecision(5) << X[i] << " ";

cout << endl;

for (int i = 0; i < A.n; i++)

X[i] = A.b[i];

X = CGM(A, X);

cout << "CGM X = ";

for (int i = 0; i < A.n; i++)

cout << fixed << setprecision(5) << X[i] << " ";

cout << endl;

delete[] X;

A.Delete();

cin.get();

cin.get();

return 0;

}

double\* ValueIterationMethod(Matrix C, double\* X)

{

int n = C.n;

double\* X0 = new double[n];

copy(X0, X, n);

double\* E = new double[n];

cout << endl;

double delta;

do

{

for (int i = 0; i < n; i++)

{

X[i] = 0;

for (int j = 0; j < n; j++)

{

X[i] += C.value[i][j] \* X0[j];

}

X[i] += C.b[i];

E[i] = abs(X[i] - X0[i]);

}

delta = Max(E,n);

cout << "X = ";

for (int i = 0; i < n; i++)

cout << setw(5) << setprecision(5) << X[i] << " ";

cout << setw(15) << " E = ";

for (int i = 0; i < n; i++)

cout << setw(5) << setprecision(5) << E[i] << " ";

cout << "delta = " << delta << endl;

copy(X0, X, n);

} while (delta > c\_delta);

delete[] E;

delete[] X0;

return X;

}

double\* VIM(Matrix A, double\* X)

{

int n = A.n;

double\* X0 = new double[n];

copy(X0, X, n);

double\* X\_1 = new double[n];

double\* E = new double[n];

double delta;

double\* oldE = new double[n];

double oldEnorm;

int k = 0;

do

{

for (int i = 0; i < n; i++)

{

X[i] = X0[i];

double sum = 0;

for (int j = 0; j < n; j++)

{

sum += A.value[i][j] \* X0[j];

}

X[i] += teta\*(A.b[i] - sum);

E[i] = -A.b[i];

oldE[i] = fabs(X[i] - X0[i]);

for (int j = 0; j < n; j++)

E[i] += A.value[i][j] \* X[j];

}

oldEnorm = Max(oldE, n);

delta = Max(E, n);

if (OUT\_VIM)

{

if (k >= 2)

{

cout << "q = " << GetQ(X\_1, X0, X, n) << " ";

}

cout << k++ << " X = ";

for (int i = 0; i < n; i++)

{

if (X[i] > 0)

cout << " ";

cout << fixed << setprecision(5) << X[i] << " ";

}

cout << "norm r = " << oldEnorm;

cout << " |A\*x-b| = " << delta << endl;

}

copy(X\_1, X0, n);

copy(X0, X, n);

} while (delta > c\_delta);

delete[] E;

delete[] X0;

delete[] X\_1;

return X;

}

double\* SOR(Matrix A, double\* X)

{

int k = 0;

int n = A.n;

double\* X0 = new double[n];

copy(X0, X, n);

double\* X\_1 = new double[n];

double\* L = new double[n];

double\* E = new double[n];

double\* oldE = new double[n];

double oldEnorm;

double delta;

do

{

for (int i = 0; i < n; i++)

{

L[i] = A.b[i];

for (int j = 0; j < i; j++)

L[i] -= A.value[i][j] \* X[j];

for (int j = i + 1; j < n; j++)

L[i] -= A.value[i][j] \* X0[j];

L[i] /= A.value[i][i];

X[i] = X0[i] + w\*(L[i] - X0[i]);

E[i] = -A.b[i];

oldE[i] = fabs(X[i] - X0[i]);

for (int j = 0; j < n; j++)

E[i] += A.value[i][j] \* X[j];

}

delta = Max(E, n);

oldEnorm = Max(oldE, n);

if (OUT\_SOR)

{

if (k >= 2)

{

cout << "q = " << fixed << GetQ(X\_1, X0, X, n) << " ";

}

cout << k++ << " X = ";

for (int i = 0; i < n; i++)

{

if (X[i] > 0)

cout << " ";

cout << fixed << setprecision(5) << X[i] << " ";

}

cout << "norm r = " << oldEnorm;

cout << " |A\*x-b| = " << delta << endl;

}

copy(X\_1, X0, n);

copy(X0, X, n);

} while (delta > c\_delta);

delete[] E;

delete[] X0;

delete[] X\_1;

return X;

}

double\* CGM(Matrix A, double\* X)

{

int n = A.n;

int k = 0;

double\* X0 = new double[n];

copy(X0, X, n);

double\* X\_1 = new double[n];

double\* E = new double[n];

double\* r = new double[n];

double\* z = new double[n];

double\* Az = new double[n];

for (int i = 0; i < n; i++)

{

r[i] = A.b[i];

for (int j = 0; j < n; j++)

r[i] -= A.value[i][j] \* X0[j];

z[i] = r[i];

}

double delta;

double\* oldE = new double[n];

double oldEnorm;

double a = 1;

double b = 1;

do

{

for (int j = 0; j < n; j++)

{

Az[j] = A.value[j][0] \* z[0];

for (int t = 1; t < n; t++)

Az[j] += A.value[j][t] \* z[t];

}

a = scalar(r, r, n) / scalar(Az, z, n);

b = 1 / scalar(r, r, n);

for (int i = 0; i < n; i++)

{

X[i] = X0[i] + a\*z[i];

E[i] = -A.b[i];

oldE[i] = fabs(X[i] - X0[i]);

for (int j = 0; j < n; j++)

E[i] += A.value[i][j] \* X[j];

r[i] -= a\*Az[i];

}

b \*= scalar(r, r, n);

for (int i = 0; i < n; i++)

{

z[i] = r[i] + b\*z[i];

}

oldEnorm = Max(oldE, n);

delta = Max(E, n);

if (OUT\_CGM)

{

if (k >= 2)

{

cout << "q = " << GetQ(X\_1, X0, X, n) << " ";

}

cout << k++ << " X = ";

for (int i = 0; i < n; i++)

{

if (X[i] > 0)

cout << " ";

cout << fixed << setprecision(5) << X[i] << " ";

}

cout << "norm r = " << oldEnorm;

cout << " |A\*x-b| = " << delta << endl;

}

copy(X\_1, X0, n);

copy(X0, X, n);

} while (delta > c\_delta);

delete[] E;

delete[] X0;

delete[] X\_1;

delete[] r;

delete[] z;

delete[] Az;

return X;

}

double\* SDM(Matrix A, double \* X)

{

int n = A.n;

int k = 0;

double\* X0 = new double[n];

copy(X0, X, n);

double\* X\_1 = new double[n];

double\* E = new double[n];

double\* r = new double[n];

double\* Ar = new double[n];

double\* oldE = new double[n];

double oldEnorm;

double delta;

double t = 1;

do

{

for (int i = 0; i < n; i++)

{

r[i] = -A.b[i];

for (int j = 0; j < n; j++)

r[i] += A.value[i][j] \* X0[j];

}

for (int i = 0; i < n; i++)

{

Ar[i] = A.value[i][0] \* r[0];

for (int j = 1; j < n; j++)

{

Ar[i] += A.value[i][j] \* r[j];

}

}

t = scalar(r, r, n) / scalar(Ar, r, n);

for (int i = 0; i < n; i++)

{

X[i] = X0[i]- t \* r[i];

oldE[i] = fabs(X[i] - X0[i]);

E[i] = -A.b[i];

for (int j = 0; j < n; j++)

E[i] += A.value[i][j] \* X[j];

}

oldEnorm = Max(oldE, n);

delta = Max(E, n);

if (OUT\_SDM)

{

if (k >= 2)

{

cout << "q = " << GetQ(X\_1, X0, X, n) << " ";

}

cout << k++ << " X = ";

for (int i = 0; i < n; i++)

{

if (X[i] > 0)

cout << " ";

cout << fixed << setprecision(5) << X[i] << " ";

}

cout << "tau = " << t;

cout << " norm r = " << oldEnorm;

cout << " |A\*x-b| = " << delta << endl;

}

copy(X\_1, X0, n);

copy(X0, X, n);

} while (delta > c\_delta);

delete[] E;

delete[] X0;

delete[] X\_1;

delete[] r;

delete[] Ar;

return X;

}

double\*\* init(int n, int m, bool fillByZero)

{

double\*\* A = new double\*[n];

for (int i = 0; i < n; i++)

A[i] = new double[m];

if (fillByZero)

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

A[i][j] = 0.0;

return A;

}

void copy(double\* A, double\* B, int n)

{

for (int i = 0; i < n; i++)

A[i] = B[i];

}

double Max(double\* E, int n)

{

double res = fabs(E[0]);

for (int i = 1; i < n; i++)

if (res < fabs(E[i]))

{

res = fabs(E[i]);

}

return res;

}

double Norm(double\*\* M, int n)

{

double sum = 0;

for (int i = 0; i < n; i++)

sum += fabs(M[0][i]);

double norm = sum;

for (int i = 1; i < n; i++)

{

sum = 0;

for (int j = 0; j < n; j++)

sum += fabs(M[i][j]);

if (sum > norm)

norm = sum;

}

return norm;

}

double scalar(double\* A, double\* B, int n)

{

double sum = A[0] \* B[0];

for (int i = 1; i < n; i++)

sum += A[i] \* B[i];

return sum;

}

double GetQ(double\* X\_1, double\* X0, double\* X, int n)

{

double\* a = new double[n];

double\* b = new double[n];

for (int i = 0; i < n; i++)

{

a[i] = X[i] - X0[0];

b[i] = X0[i] - X\_1[0];

}

double res = Max(a, n) / Max(b, n);

delete[] a;

delete[] b;

return res;

}