**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

Лабораторная работа № 1

Выполнил: Ульяницкий Владимир Александрович

2 курс, 3 группа

Преподаватель: Полещук Максим Александрович

Минск, 2018

1. **Постановка задачи.**

**Задание 1**. Пусть дана система линейных алгебраических уравнений вида

*a11x1 + a12x2 +  … + a1nxn = b1 ,*

*a21x2 + a22x2 +  … + a2nxn = b2 ,*

*. . . . . . . . . . . . . . . . . . . . . . . .*

*an1x1 + an2x2 +  … + annxn = bn .*

Разработать программу численного решения СЛАУ методом Гаусса с выбором ведущего элемента по столбцу (алгоритм GEPP).

Задать хорошо обусловленную матрицу (можно показать, что верхняя оценка  не является величиной, намного большей 1) *A* с диагональным преобладанием следующим образом:

1. *n* — размер квадратной матрицы, одно из чисел в пределах от 20 до 25;
2. недиагональные элементы **, *i≠j*, выбираются из чисел 0, –0,01, –0,02, –0,03 произвольным образом;
3. **

Правую часть *b* задать умножением матрицы *A* на вектор ***.*: **, где *s* — номер в списке академической группы.

Нижний треугольник (под главной диагональю) матрицы *L* и верхний треугольник матрицы *U* хранить на месте матрицы *A*. Операцию обмена строк производить обменом указателями на строки матрицы *A*.

Вычислить и представить в отчёте следующие величины:

1. Вектор приближённого решения .
2. Относительная погрешность решения в нормах , где  — точное решение.
3. **Входные данные**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.37 | -0.01 | -0.03 | -0.01 | -0.03 | -0.02 | -0 | -0.03 | -0.03 | -0.03 | -0.03 | -0 | -0 | -0.03 | -0.03 | -0 | -0.03 | -0.02 | -0 | -0.01 | -0 | -0.02 | -0 | -3.35 |
| -0.02 | 0.25 | -0 | -0.01 | -0.02 | -0.03 | -0.01 | -0.01 | -0 | -0 | -0 | -0.01 | -0.03 | -0.01 | -0.02 | -0.02 | -0 | -0.01 | -0.01 | -0 | -0.01 | -0.02 | -0 | -2.06 |
| -0.03 | -0.02 | 0.39 | -0 | -0.02 | -0.03 | -0 | -0.03 | -0.02 | -0 | -0.01 | -0.02 | -0.02 | -0.02 | -0.01 | -0.01 | -0.02 | -0.03 | -0.01 | -0.03 | -0 | -0.03 | -0.02 | -3.37 |
| -0.01 | -0.02 | -0.03 | 0.4 | -0.01 | -0.01 | -0.02 | -0.03 | -0 | -0.02 | -0 | -0.01 | -0.03 | -0.02 | -0 | -0.01 | -0.03 | -0.02 | -0.02 | -0.01 | -0.03 | -0.03 | -0.03 | -3.39 |
| -0 | -0.02 | -0.01 | -0.02 | 0.29 | -0.03 | -0 | -0 | -0.03 | -0.01 | -0 | -0 | -0.03 | -0 | -0.01 | -0.01 | -0.01 | -0.02 | -0 | -0.03 | -0.03 | -0.01 | -0.01 | -1.94 |
| -0 | -0.03 | -0.01 | -0.02 | -0 | 0.38 | -0.03 | -0.03 | -0 | -0 | -0.02 | -0.03 | -0.03 | -0.02 | -0.02 | -0.01 | -0 | -0.03 | -0.03 | -0.01 | -0.01 | -0.02 | -0.02 | -2.25 |
| -0.01 | -0.01 | -0.01 | -0.02 | -0.03 | -0 | 0.3 | -0.02 | -0.01 | -0.03 | -0.01 | -0 | -0.02 | -0 | -0.03 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0 | -0.03 | -0.01 | -1.13 |
| -0 | -0 | -0.01 | -0.02 | -0.01 | -0 | -0.01 | 0.3 | -0 | -0.01 | -0.02 | -0.01 | -0.01 | -0.02 | -0.01 | -0.01 | -0.02 | -0.02 | -0.02 | -0.03 | -0.03 | -0.02 | -0.01 | -1.75 |
| -0.02 | -0.03 | -0.02 | -0 | -0.03 | -0.02 | -0 | -0 | 0.3 | -0 | -0 | -0 | -0.03 | -0 | -0.02 | -0.02 | -0 | -0 | -0.03 | -0.02 | -0.03 | -0.02 | -0 | -0.54 |
| -0 | -0.03 | -0.03 | -0 | -0 | -0.03 | -0.01 | -0.03 | -0.03 | 0.41 | -0.01 | -0.03 | -0.01 | -0.03 | -0.01 | -0.01 | -0.02 | -0.02 | -0.02 | -0.03 | -0.03 | -0.02 | -0 | -0.67 |
| -0 | -0 | -0.01 | -0.01 | -0.02 | -0 | -0 | -0.01 | -0.03 | -0 | 0.28 | -0.02 | -0 | -0.03 | -0.03 | -0 | -0.03 | -0.01 | -0.01 | -0.02 | -0.01 | -0 | -0.03 | -0.51 |
| -0.03 | -0.03 | -0.03 | -0.02 | -0.02 | -0.02 | -0.02 | -0 | -0.01 | -0.01 | -0.02 | 0.43 | -0.03 | -0.03 | -0.02 | -0.02 | -0.03 | -0.01 | -0.01 | -0.01 | -0.03 | -0 | -0.02 | 0.75 |
| -0 | -0.03 | -0.01 | -0.02 | -0 | -0 | -0 | -0.02 | -0.03 | -0.03 | -0.02 | -0.03 | 0.38 | -0.01 | -0.03 | -0.02 | -0.01 | -0.03 | -0.01 | -0.02 | -0.01 | -0.03 | -0.01 | 0.37 |
| -0.03 | -0 | -0.02 | -0.02 | -0.01 | -0.02 | -0.02 | -0.02 | -0.03 | -0.03 | -0.02 | -0.03 | -0.03 | 0.41 | -0.01 | -0 | -0.02 | -0.01 | -0.02 | -0.02 | -0.02 | -0.02 | -0 | 1.47 |
| -0.03 | -0.03 | -0.02 | -0.02 | -0 | -0.03 | -0.01 | -0 | -0.01 | -0 | -0.02 | -0.02 | -0.01 | -0 | 0.34 | -0.01 | -0.01 | -0.03 | -0.01 | -0.03 | -0.02 | -0.01 | -0.01 | 1.63 |
| -0.03 | -0.03 | -0.02 | -0.01 | -0 | -0.03 | -0.03 | -0 | -0 | -0.02 | -0.01 | -0.02 | -0.03 | -0.01 | -0.03 | 0.36 | -0.02 | -0.01 | -0.03 | -0.01 | -0.01 | -0 | -0 | 2.37 |
| -0.03 | -0 | -0.01 | -0 | -0 | -0 | -0 | -0 | -0.01 | -0.03 | -0.01 | -0 | -0.03 | -0.02 | -0.02 | -0.01 | 0.25 | -0 | -0.01 | -0.03 | -0.01 | -0 | -0.02 | 1.32 |
| -0.01 | -0.03 | -0.02 | -0.02 | -0.02 | -0.01 | -0 | -0 | -0 | -0.02 | -0.01 | -0.03 | -0 | -0.01 | -0.01 | -0.02 | -0 | 0.3 | -0.03 | -0.02 | -0.02 | -0 | -0.01 | 2.35 |
| -0.02 | -0 | -0.02 | -0.02 | -0.03 | -0 | -0.01 | -0.02 | -0.02 | -0.02 | -0.03 | -0.03 | -0.03 | -0.02 | -0.01 | -0.02 | -0.01 | -0.01 | 0.38 | -0.02 | -0.01 | -0.01 | -0.01 | 3.28 |
| -0.02 | -0 | -0.03 | -0.01 | -0.02 | -0.02 | -0.01 | -0.02 | -0.01 | -0.01 | -0.02 | -0.01 | -0.01 | -0 | -0.02 | -0.03 | -0.02 | -0 | -0.01 | 0.32 | -0.01 | -0.02 | -0.01 | 3.17 |
| -0.01 | -0.03 | -0.02 | -0.03 | -0.03 | -0.03 | -0.03 | -0 | -0 | -0.02 | -0.01 | -0.03 | -0 | -0.03 | -0 | -0.02 | -0 | -0.03 | -0 | -0 | 0.37 | -0.01 | -0.03 | 4.34 |
| -0.02 | -0.02 | -0 | -0 | -0.03 | -0.03 | -0.03 | -0 | -0.01 | -0.01 | -0.03 | -0.01 | -0 | -0.03 | -0 | -0.03 | -0.03 | -0 | -0.02 | -0.01 | -0.03 | 0.35 | -0 | 4.06 |
| -0.02 | -0.01 | -0.03 | -0.01 | -0 | -0 | -0 | -0.03 | -0.02 | -0 | -0.02 | -0.01 | -0.01 | -0 | -0 | -0 | -0.02 | -0.03 | -0.01 | -0 | -0.03 | -0 | 0.26 | 3.44 |

**3. Ход работы**

1.Создание матрицы, соответствующей условию поставленной задачи.

2.Умножение матрицы на заданный столбец *Х*. Получение столбца свободных членов. Запись расширенной матрицы.

3.Поиск решения с учетом возможных ошибок деления на 0 или бесконечного числа решений.

4.Получение *LU*-разложения *A=LU* по формулам прямого хода с выбором ведущего элемента по столбцу

, ;

5.Решение системы *Ly=b* с нижней треугольной матрицей (заменяет преобразование по формулам вектора правой части системы уравнений)

;

6.Решение системы *Ux=y* с верхней треугольной матрицей по формулам обратного хода с учетом обозначений

.

**4. Вывод программы**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.37 | -0.01 | -0.03 | -0.01 | -0.03 | -0.02 | -0 | -0.03 | -0.03 | -0.03 | -0.03 | -0 | -0 | -0.03 | -0.03 | -0 | -0.03 | -0.02 | -0 | -0.01 | -0 | -0.02 | -0 | -3.35 |
| -0.054054 | 0.24946 | -0.0016216 | -0.010541 | -0.021622 | -0.031081 | -0.01 | -0.011622 | -0.0016216 | -0.0016216 | -0.0016216 | -0.01 | -0.03 | -0.011622 | -0.021622 | -0.02 | -0.0016216 | -0.011081 | -0.01 | -0.00054054 | -0.01 | -0.021081 | -0 | -2.06 |
| -0.081081 | -0.083424 | 0.38743 | -0.0016901 | -0.024236 | -0.034215 | -0.00083424 | -0.033402 | -0.022568 | -0.0025677 | -0.012568 | -0.020834 | -0.022503 | -0.023402 | -0.014236 | -0.011668 | -0.022568 | -0.032546 | -0.010834 | -0.030856 | -0.00083424 | -0.03338 | -0.02 | -3.37 |
| -0.027027 | -0.081257 | -0.079866 | 0.39874 | -0.014503 | -0.015799 | -0.020879 | -0.034423 | -0.002745 | -0.021148 | -0.0019463 | -0.012477 | -0.034235 | -0.023624 | -0.0037047 | -0.012557 | -0.032745 | -0.02404 | -0.021678 | -0.012779 | -0.030879 | -0.034919 | -0.031597 | -3.39 |
| -0 | -0.080173 | -0.026147 | -0.052388 | 0.28687 | -0.034214 | -0.0019174 | -0.0036084 | -0.030864 | -0.011305 | -0.00056058 | -0.0020001 | -0.034787 | -0.0027813 | -0.0123 | -0.012566 | -0.012436 | -0.022999 | -0.0022207 | -0.03152 | -0.032441 | -0.014392 | -0.012178 | -1.94 |
| -0 | -0.12026 | -0.026314 | -0.053449 | -0.013989 | 0.37404 | -0.032367 | -0.034167 | -0.0013674 | -0.0015511 | -0.020638 | -0.032446 | -0.036516 | -0.023315 | -0.023345 | -0.013559 | -0.002713 | -0.033796 | -0.032677 | -0.012001 | -0.013329 | -0.025481 | -0.022385 | -2.25 |
| -0.027027 | -0.04117 | -0.028076 | -0.052043 | -0.11551 | -0.020198 | 0.2976 | -0.025125 | -0.015247 | -0.033387 | -0.011813 | -0.0025323 | -0.028404 | -0.003968 | -0.034186 | -0.01353 | -0.014707 | -0.016501 | -0.012761 | -0.015707 | -0.0060586 | -0.03634 | -0.014065 | -1.13 |
| -0 | -0 | -0.025811 | -0.050268 | -0.039581 | -0.0081047 | -0.038337 | 0.29602 | -0.0025377 | -0.012869 | -0.021065 | -0.011604 | -0.015064 | -0.022243 | -0.01254 | -0.012058 | -0.023307 | -0.023865 | -0.022211 | -0.033386 | -0.033198 | -0.024786 | -0.013307 | -1.75 |
| -0.054054 | -0.12243 | -0.05632 | -0.0048307 | -0.12446 | -0.083274 | -0.014469 | -0.029558 | 0.29264 | -0.0044666 | -0.0051193 | -0.0057884 | -0.043332 | -0.0074791 | -0.029429 | -0.026412 | -0.0059247 | -0.011008 | -0.035778 | -0.028543 | -0.037637 | -0.030882 | -0.0052557 | -0.54 |
| -0 | -0.12026 | -0.077936 | -0.0035094 | -0.015826 | -0.098924 | -0.048968 | -0.13103 | -0.11504 | 0.40536 | -0.017159 | -0.038422 | -0.027994 | -0.039624 | -0.022929 | -0.02018 | -0.02699 | -0.032862 | -0.033042 | -0.042628 | -0.042184 | -0.036588 | -0.0071138 | -0.67 |
| -0 | -0 | -0.025811 | -0.025189 | -0.073171 | -0.010118 | -0.0034114 | -0.041972 | -0.11305 | -0.006416 | 0.27776 | -0.022723 | -0.010165 | -0.033685 | -0.035714 | -0.0053416 | -0.034216 | -0.015984 | -0.016551 | -0.028501 | -0.019247 | -0.0079422 | -0.033676 | -0.51 |
| -0.081081 | -0.12351 | -0.084228 | -0.055814 | -0.097443 | -0.087044 | -0.0856 | -0.04756 | -0.065742 | -0.046945 | -0.10371 | 0.41799 | -0.052445 | -0.046701 | -0.039977 | -0.031528 | -0.045381 | -0.028728 | -0.024187 | -0.028071 | -0.045898 | -0.021467 | -0.032592 | 0.75 |
| -0 | -0.12026 | -0.026314 | -0.053449 | -0.013989 | -0.015937 | -0.0096881 | -0.0843 | -0.1085 | -0.082565 | -0.089389 | -0.095212 | 0.35844 | -0.027141 | -0.04719 | -0.032934 | -0.025114 | -0.044578 | -0.025586 | -0.036995 | -0.030056 | -0.047454 | -0.021271 | 0.37 |
| -0.081081 | -0.0032503 | -0.057914 | -0.052523 | -0.051131 | -0.070269 | -0.079132 | -0.104 | -0.12655 | -0.096037 | -0.10861 | -0.10186 | -0.15616 | 0.38312 | -0.041047 | -0.019535 | -0.045481 | -0.038316 | -0.043486 | -0.049879 | -0.048503 | -0.051319 | -0.019158 | 1.47 |
| -0.081081 | -0.12351 | -0.058417 | -0.055704 | -0.02554 | -0.10484 | -0.05339 | -0.043078 | -0.054533 | -0.01725 | -0.099965 | -0.072933 | -0.087869 | -0.053471 | 0.3139 | -0.025396 | -0.030324 | -0.053096 | -0.028709 | -0.050838 | -0.04027 | -0.035756 | -0.025957 | 1.63 |
| -0.081081 | -0.12351 | -0.058417 | -0.030625 | -0.024272 | -0.10366 | -0.1187 | -0.045554 | -0.023412 | -0.070355 | -0.069369 | -0.075339 | -0.14618 | -0.085384 | -0.19621 | 0.33635 | -0.049678 | -0.048783 | -0.057718 | -0.045866 | -0.041708 | -0.039478 | -0.022281 | 2.37 |
| -0.081081 | -0.0032503 | -0.032103 | -0.0022554 | -0.01155 | -0.0086939 | -0.0013774 | -0.01349 | -0.046445 | -0.081751 | -0.053904 | -0.013947 | -0.10445 | -0.085361 | -0.12006 | -0.067408 | 0.22781 | -0.025711 | -0.030371 | -0.056828 | -0.032876 | -0.025057 | -0.032818 | 1.32 |
| -0.027027 | -0.12134 | -0.054223 | -0.054274 | -0.089014 | -0.053658 | -0.014446 | -0.028437 | -0.018771 | -0.059986 | -0.053452 | -0.093147 | -0.057288 | -0.070952 | -0.098061 | -0.10946 | -0.11852 | 0.26569 | -0.05882 | -0.056959 | -0.053684 | -0.033038 | -0.032479 | 2.35 |
| -0.054054 | -0.0021668 | -0.055817 | -0.051808 | -0.11773 | -0.021133 | -0.040523 | -0.092761 | -0.094116 | -0.067088 | -0.13319 | -0.095889 | -0.14744 | -0.11367 | -0.13274 | -0.12725 | -0.21666 | -0.21961 | 0.32525 | -0.091187 | -0.071941 | -0.064565 | -0.050884 | 3.28 |
| -0.054054 | -0.0021668 | -0.081628 | -0.026838 | -0.083786 | -0.072805 | -0.044244 | -0.098636 | -0.058608 | -0.040632 | -0.10027 | -0.047902 | -0.079078 | -0.047766 | -0.13461 | -0.13834 | -0.22254 | -0.15736 | -0.1692 | 0.24595 | -0.070018 | -0.075201 | -0.052798 | 3.17 |
| -0.027027 | -0.12134 | -0.054223 | -0.079353 | -0.12514 | -0.11149 | -0.12353 | -0.047727 | -0.028938 | -0.072233 | -0.065044 | -0.10165 | -0.082295 | -0.13603 | -0.10139 | -0.12815 | -0.15922 | -0.28846 | -0.17577 | -0.33673 | 0.27292 | -0.10399 | -0.097313 | 4.34 |
| -0.054054 | -0.08234 | -0.0045302 | -0.0035515 | -0.117 | -0.1012 | -0.11559 | -0.032553 | -0.059668 | -0.044075 | -0.13352 | -0.048382 | -0.061229 | -0.12332 | -0.092581 | -0.14278 | -0.27003 | -0.16157 | -0.21282 | -0.36978 | -0.45074 | 0.21338 | -0.11023 | 4.06 |
| -0.054054 | -0.042254 | -0.081795 | -0.027898 | -0.017158 | -0.016631 | -0.0055257 | -0.12355 | -0.083934 | -0.011988 | -0.095147 | -0.042118 | -0.068168 | -0.041805 | -0.06021 | -0.038773 | -0.18007 | -0.21561 | -0.14445 | -0.24529 | -0.35496 | -0.46717 | 0.12615 | 3.44 |

Полученное разложение:

На месте нижнего треугольника получена матрица *L*, на месте верхнего треугольника хранятся элементы матрицы *U*.

Полученный вектор y:

y = (-3.35, -2.2411, -3.8286, -3.9684, -2.4277, -2.8663, -1.9651, -2.243, -1.8658, -2.1786, -1.2412, -1.457, -1.1321, -0.85872, -0.44223, 0.066508, 0.23117, 0.60024, 1.2811, 1.6344, 2.7808, 4.4283, 5.6766)

Вектор приближенного решения *x\**, с точностью до 17 знаков после запятой:

x\* = (23.000001907348633, 24.000001907348633, 25, 26, 27.000001907348633, 28, 29, 29.999998092651367, 31.000003814697266, 31.999998092651367, 33, 34, 35.000003814697266, 36.000003814697266, 37, 37.999996185302734, 39, 40.000007629394531, 41, 42, 43, 44.000003814697266, 45)

Относительная погрешность в нормах:

Relative error in norms: 1.6954210479980247e-07

**5. Листинг**

// Gauss.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

#include <fstream>

#include <cmath>

#include <iomanip>

#include <ctime>

using namespace std;

void matrixOut(float \*\* A, int n)

{

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

{

for (int j = 0; j < n + 1; j++)

cout << setprecision(5) << A[i][j] << " ";

cout << endl;

}

cout << endl;

}

int main()

{

//n — размер квадратной матрицы, одно из чисел в пределах от 20 до 25;

srand(time(NULL));

int n = rand() % 6 + 20;

int m = n + 1;

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

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

A[i] = new float[m];

//вектор X

float\* x = new float[n];

int s = 23;

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

x[i] = s + i;

}

//заполняем матрицу элементами, кроме диагональных

srand(time(NULL));

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

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

if (i != j)

A[i][j] = rand() % 4 \* -0.01;

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

A[i][m - 1] = 0;

//задаем диагональные элементы

float sum;

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

{

sum = 0;

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

if (i != j)

sum += A[i][j];

A[i][i] = -1\*sum + 0.01;

}

//задаем вектор b = A \* x

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

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

A[i][m - 1] += A[i][j] \* x[j];

cout <<"Input: " << endl;

matrixOut(A, n);

//вектор приближённого решения

float \*approxX = new float[n];

bool stop = false;

for (int k = 0; k < n - 1; k++) {

if (stop)

break;

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

{

//макс элемент в столбце

float maxCol = A[k][k];

int maxColIndex = k;

for (int h = i; h < n; h++) {

if (A[h][k] > A[k][k]) {

maxCol = A[h][k];

maxColIndex = h;

}

}

if ((maxCol == 0) && (A[k][m - 1] != 0))

{

cout << "infinitely many solutions" << endl; //бесконечно много решений

stop = true;

break;

}

else

if ((maxCol == 0) && (A[k][m - 1] == 0))

{

stop = true;

cout << "no solutions" << endl; //нет решений

break;

}

//меняем строки, если макс элемент не первый в столбце

if (maxCol != A[k][k])

{

float\* temp = A[k];

A[k] = A[maxColIndex];

A[maxColIndex] = temp;

}

//проверка, что диагональные элементы не 0

if (A[k][k] == 0)

{

int indexNotNull = -1;

for (int m = k; k < n; m++)

if (A[m][k] != 0)

{

indexNotNull = m;

break;

}

if (indexNotNull == -1)

break;

else

{

//меняем строки с ненулемым элементом

float\* temp = A[k];

A[k] = A[indexNotNull];

A[indexNotNull] = temp;

}

}

float Lik = A[i][k] / A[k][k];

A[i][k] = Lik;

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

A[i][j] = A[i][j] - Lik\* A[k][j];

}

}

cout << endl << "LU: " << endl;

matrixOut(A, n);

// L \* y = b

float \*y = new float[n];

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

float sum = 0;

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

sum += y[j] \* A[i][j];

}

y[i] = A[i][m - 1] - sum;

}

cout << "y = (" << endl;

for (int k = 0; k < n; k++) {

cout << y[k];

if (k != n - 1)

cout << ", ";

}

cout << ") " << endl;

// Обратный ход

for (int i = n - 1; i >= 0; i--) {

float sum = 0;

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

sum += approxX[j] \* A[i][j];

}

approxX[i] = 1 / A[i][i] \* (y[i] - sum);

}

cout << "x\* = (";

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

cout << setprecision(17) << approxX[i];

if (i != n - 1) {

cout << ", ";

}

}

cout << ") " << endl;

//Относительная погрешность решения

float approxNorm = 0;

float diffNorm = 0;

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

{

if (fabs(approxX[i]) > approxNorm)

approxNorm = fabs(approxX[i]);

if (fabs(approxX[i] - x[i]) > diffNorm)

diffNorm = fabs(approxX[i] - x[i]);

}

cout << "Relative error in norms: " << diffNorm / approxNorm << endl;

system("pause");

return 0;

}

**1. Постановка задачи.**

**Задание 2.** Пусть дана система линейных алгебраических уравнений вида

*a11x1 + a12x2 +  … + a1nxn = b1 ,*

*a21x2 + a22x2 +  … + a2nxn = b2 ,*

*. . . . . . . . . . . . . . . . . . . . . . . .*

*an1x1 + an2x2 +  … + annxn = bn .*

с положительно определенной симметрической матрицей. Разработать программу численного решения СЛАУ методом Холецкого.

Матрицу системы задать **, где ** — симметрическая матрица:

1. *n* — размер квадратной матрицы, одно из чисел в пределах от 20 до 25;
2. недиагональные элементы **, *i<j*, выбираются из чисел 0, –1, –2, –3, *–*4 произвольным образом; если *i>j*, то полагается **;
3. **

Правую часть *d* задать умножением матрицы *A* на вектор **: **, где *s* — номер в списке академической группы.

В процессе факторизации матрицы *A* (*A=LDLT*) нижнюю треугольную матрицу *L* хранить на месте нижнего треугольника матрицы *A*. Диагональную матрицу *D* хранить в виде отдельного вектора *d*.

Вычислить и представить в отчете величины из задания 1.

**2**. **Входные данные**

Матрица *A:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2643 | 13 | -25 | -301 | -300 | 77 | -97 | 103 | -294 | -95 | -109 | -18 | -193 | -119 | -294 | -216 | -84 | -78 | 81 | -89 | -299 | -7 | -199 | -99 | -32992 |
| 13 | 3925 | -324 | -123 | -217 | -109 | 0 | -326 | -102 | 20 | -200 | -200 | -326 | 94 | -319 | -225 | -225 | 146 | -336 | -335 | -330 | -307 | 112 | -305 | -47260 |
| -25 | -324 | 1469 | 81 | 90 | -76 | -88 | 87 | -2 | -207 | 58 | -22 | 85 | -92 | -8 | -102 | 1 | -211 | -84 | -303 | 83 | -277 | -191 | 59 | -18260 |
| -301 | -123 | 81 | 2055 | 113 | 0 | -98 | -1 | -4 | -205 | -93 | -111 | -5 | -19 | 92 | -202 | -195 | -307 | -91 | 96 | 0 | -211 | -287 | -183 | -23100 |
| -300 | -217 | 90 | 113 | 2867 | -20 | 14 | 2 | -306 | -317 | 112 | -113 | -291 | 74 | -70 | -105 | 15 | -328 | -108 | -216 | -293 | -95 | -209 | -298 | -29409 |
| 77 | -109 | -76 | 0 | -20 | 1971 | -196 | 112 | 89 | -109 | -197 | -183 | -8 | -269 | 52 | -313 | -3 | 0 | -271 | -100 | 80 | -307 | -201 | -18 | -19176 |
| -97 | 0 | -88 | -98 | 14 | -196 | 1959 | -24 | -195 | -289 | -4 | 72 | -196 | 78 | 92 | -92 | -103 | -207 | -98 | -319 | -188 | 2 | -86 | 64 | -11754 |
| 103 | -326 | 87 | -1 | 2 | 112 | -24 | 2455 | -111 | -316 | -284 | -276 | 105 | -297 | -309 | -327 | 114 | 0 | -196 | 20 | -96 | -226 | -114 | -94 | -16623 |
| -294 | -102 | -2 | -4 | -306 | 89 | -195 | -111 | 2545 | 103 | 67 | 77 | -93 | -185 | -178 | -208 | -292 | -86 | -106 | -312 | -89 | -207 | -197 | 87 | -11580 |
| -95 | 20 | -207 | -205 | -317 | -109 | -289 | -316 | 103 | 2653 | -295 | 98 | -97 | 66 | -108 | 19 | 101 | -95 | -204 | -118 | -304 | 105 | -302 | -103 | -4250 |
| -109 | -200 | 58 | -93 | 112 | -197 | -4 | -284 | 67 | -295 | 1987 | -256 | -292 | 68 | 91 | -303 | -27 | -13 | 83 | -108 | -184 | 95 | 75 | -270 | -1301 |
| -18 | -200 | -22 | -111 | -113 | -183 | 72 | -276 | 77 | 98 | -256 | 1709 | -8 | -171 | 70 | -89 | 85 | -124 | -104 | -212 | -281 | -2 | 71 | -11 | 641 |
| -193 | -326 | 85 | -5 | -291 | -8 | -196 | 105 | -93 | -97 | -292 | -8 | 2343 | -36 | -183 | 7 | 111 | -217 | -20 | -318 | 21 | -199 | -17 | -172 | 2369 |
| -119 | 94 | -92 | -19 | 74 | -269 | 78 | -297 | -185 | 66 | 68 | -171 | -36 | 1401 | 83 | 16 | -288 | -111 | -87 | -317 | 71 | 90 | -93 | 44 | 2504 |
| -294 | -319 | -8 | 92 | -70 | 52 | 92 | -309 | -178 | -108 | 91 | 70 | -183 | 83 | 2075 | 15 | -294 | -316 | 84 | -322 | 15 | 14 | -104 | -177 | 5328 |
| -216 | -225 | -102 | -202 | -105 | -313 | -92 | -327 | -208 | 19 | -303 | -89 | 7 | 16 | 15 | 3185 | -209 | -337 | -94 | -351 | 15 | 124 | -203 | -4 | 17849 |
| -84 | -225 | 1 | -195 | 15 | -3 | -103 | 114 | -292 | 101 | -27 | 85 | 111 | -288 | -294 | -209 | 2753 | -316 | -206 | -214 | -216 | -100 | -199 | -208 | 4335 |
| -78 | 146 | -211 | -307 | -328 | 0 | -207 | 0 | -86 | -95 | -13 | -124 | -217 | -111 | -316 | -337 | -316 | 3299 | -102 | 27 | -109 | -111 | -301 | -102 | 15934 |
| 81 | -336 | -84 | -91 | -108 | -271 | -98 | -196 | -106 | -204 | 83 | -104 | -20 | -87 | 84 | -94 | -206 | -102 | 2053 | 108 | 86 | -94 | -282 | -11 | 18161 |
| -89 | -335 | -303 | 96 | -216 | -100 | -319 | 20 | -312 | -118 | -108 | -212 | -318 | -317 | -322 | -351 | -214 | 27 | 108 | 3805 | -313 | -330 | 116 | 106 | 34626 |
| -299 | -330 | 83 | 0 | -293 | 80 | -188 | -96 | -89 | -304 | -184 | -281 | 21 | 71 | 15 | 15 | -216 | -109 | 86 | -313 | 2353 | -109 | -11 | 99 | 27616 |
| -7 | -307 | -277 | -211 | -95 | -307 | 2 | -226 | -207 | 105 | 95 | -2 | -199 | 90 | 14 | 124 | -100 | -111 | -94 | -330 | -109 | 2445 | -6 | -286 | 27105 |
| -199 | 112 | -191 | -287 | -209 | -201 | -86 | -114 | -197 | -302 | 75 | 71 | -17 | -93 | -104 | -203 | -199 | -301 | -282 | 116 | -11 | -6 | 2547 | 82 | 33461 |
| -99 | -305 | 59 | -183 | -298 | -18 | 64 | -94 | 87 | -103 | -270 | -11 | -172 | 44 | -177 | -4 | -208 | -102 | -11 | 106 | 99 | -286 | 82 | 1801 | 26604 |

**3. Ход работы**

1. Создание матрицы *B*, соответствующей условию поставленной задачи. получение матрицы *А*.

2. Умножение матрицы на заданный столбец *x*. Получение столбца свободных членов *d*. Запись расширенной матрицы.

3. За основу решения *LDLT* разложения берется алгоритм разложения Гаусса.

Элементы матрицы L – это вычисленные lik, а элементы главной диагонали  
диагональной матрицы D – это вычисленные d11, d22, …, dnn.

Вычисленные коэффициенты *l* будем записывать в нижний треугольник нашей матрицы *А*, а диагональные *d* соответсвенно на диагональ нашей матрицы.

Т.о. общий ход алгоритма следующий:

Осуществляются преобразования прямого хода метода Гаусса, но на место нулевых элементов (ниже ведущего элемента) на каждом *k*-м шаге записываются величины (на которые умножается *k*-е уравнение перед вычитанием его из *i*-го уравнения), которые и есть элементы *lik* матрицы *L*. Вместо элементов верхнего треугольника используются равные им вычисленные ранее элементы нижнего треугольника; для хранения вычисленных на шаге *k*-1 величин (они еще потребуются, но на текущем шаге *k* элементы *aik* переопределяются), используется временный массив *ti.*

После завершения вычислений на месте элементов нижнего треугольника матрицы *А* расположены соответствующие элементы матриц *L* и *D*.

Далее решение идет по алгоритму:

*L \* y = d*

*DLT \* x = y*

**4. Вывод программы**

Полученный вектор *y*:

y = (-32992, -47098, -22458, -27255, -33727, -20876, -17738, -17375, -22967, -21235, -11822, -10522, -14576, -6307.9, -9303.4, -11294, -9026.7, -9695.9, -5173, -5453.2, -2668.1, -2003.8, -285.88, 1089.1)

Вектор приближенного решения *x\** с точностью до 17 знаков после запятой:

x\*: (23.00040435791015625, 24.0004024505615234375, 25.0004062652587890625, 26.0004062652587890625, 27.000400543212890625, 28.0004024505615234375, 29.000400543212890625, 30.000396728515625, 31.0004119873046875, 32.000408172607421875, 33.000396728515625, 34.000400543212890625, 35.00040435791015625, 36.000408172607421875, 37.000408172607421875, 38.000396728515625, 39.000400543212890625, 40.000400543212890625, 41.0004119873046875, 42.000408172607421875, 43.0004119873046875, 44.000396728515625, 45.000408172607421875, 46.000400543212890625)

Относительная погрешность в нормах:

Relative error in norms: 8.95616813068045303225517273e-06

Матрица после преобразований (*LDLT*-разложение):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2643,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -32992,00 |
| 0.0049187 | 3924.9 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -47260,00 |
| 0.0094589 | 0.082518 | 1442,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -18260,00 |
| 0.11389 | 0.030961 | 0.047242 | 2013.7 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -23100,00 |
| 0.11351 | 0.054912 | 0.048111 | 0.034207 | 2815.4 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -29409,00 |
| 0.029134 | 0.027868 | 0.058457 | 0.0046506 | 0.004806 | 1960.7 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -19176,00 |
| 0.036701 | 0.00012156 | 0.061634 | 0.052059 | 0.0038637 | 0.10089 | 1924.5 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -11754,00 |
| 0.038971 | 0.083188 | 0.042323 | 0.0011233 | 0.0025206 | 0.05276 | 0.0031552 | 2415.7 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -16623,00 |
| 0.11124 | 0.025619 | 0.0090694 | 0.019853 | 0.12179 | 0.047196 | 0.10289 | 0.047095 | 2436.9 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -11580,00 |
| 0.035944 | 0.0052147 | 0.143 | 0.10202 | 0.11001 | 0.060311 | 0.16972 | 0.12322 | 0.0029728 | 2465.7 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -4250,00 |
| 0.041241 | 0.05082 | 0.028092 | 0.056364 | 0.032183 | 0.099992 | 0.016377 | 0.1191 | 0.021082 | 0.14052 | 1857.8 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | -1301,00 |
| 0.0068104 | 0.050934 | 0.026814 | 0.058306 | 0.042674 | 0.097078 | 0.02293 | 0.11613 | 0.021696 | 0.011743 | 0.17082 | 1577.9 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 641,00 |
| 0.073023 | 0.082817 | 0.039078 | 0.019718 | 0.11796 | 0.0048723 | 0.10461 | 0.033978 | 0.073829 | 0.064429 | 0.17366 | 0.048776 | 2152.6 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 2369,00 |
| 0.045025 | 0.024099 | 0.059167 | 0.012709 | 0.0251 | 0.1364 | 0.020668 | 0.11032 | 0.075638 | 0.005079 | 0.007749 | 0.13896 | 0.016155 | 1274.4 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 2504,00 |
| 0.11124 | 0.080907 | 0.025648 | 0.025044 | 0.04289 | 0.024858 | 0.045195 | 0.13451 | 0.099114 | 0.061629 | 0.011757 | 0.015448 | 0.11374 | 0.023281 | 1897.9 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 5328,00 |
| 0.081725 | 0.057055 | 0.084964 | 0.1131 | 0.04551 | 0.16304 | 0.078495 | 0.1309 | 0.11258 | 0.050904 | 0.22205 | 0.15922 | 0.074785 | 0.098126 | 0.052392 | 2808.8 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 17849,00 |
| 0.031782 | 0.057221 | 0.012709 | 0.10461 | 0.00043351 | 0.0035178 | 0.061757 | 0.041032 | 0.13071 | 0.027476 | 0.015003 | 0.04765 | 0.021747 | 0.22814 | 0.1654 | 0.11378 | 2500.7 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 4335,00 |
| 0.029512 | 0.037296 | 0.13846 | 0.14993 | 0.10971 | 0.0027542 | 0.12328 | 0.0091425 | 0.065437 | 0.092994 | 0.022015 | 0.092886 | 0.14473 | 0.12187 | 0.19433 | 0.19108 | 0.20737 | 2762.9 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 15934,00 |
| 0.030647 | 0.085708 | 0.076969 | 0.043177 | 0.038703 | 0.14757 | 0.070213 | 0.086092 | 0.054024 | 0.12235 | 0.013222 | 0.11421 | 0.047786 | 0.13712 | 0.0050454 | 0.1084 | 0.12211 | 0.12767 | 1725.9 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 18161,00 |
| 0.033674 | 0.08524 | 0.22985 | 0.045271 | 0.082277 | 0.065102 | 0.18181 | 0.0061322 | 0.15969 | 0.10023 | 0.080753 | 0.16479 | 0.22572 | 0.3179 | 0.23846 | 0.24316 | 0.20857 | 0.21232 | 0.21013 | 2638.2 | 0,00 | 0,00 | 0,00 | 0,00 | 34626,00 |
| 0.11313 | 0.083703 | 0.036797 | 0.023206 | 0.12287 | 0.041421 | 0.097981 | 0.049561 | 0.082943 | 0.15798 | 0.14303 | 0.22432 | 0.088512 | 0.011719 | 0.065448 | 0.085149 | 0.11911 | 0.14863 | 0.10403 | 0.3251 | 1583.1 | 0,00 | 0,00 | 0,00 | 27616,00 |
| 0.0026485 | 0.078209 | 0.2097 | 0.1028 | 0.032329 | 0.16959 | 0.031608 | 0.091722 | 0.09626 | 0.010143 | 0.010585 | 0.056795 | 0.11534 | 0.018776 | 0.046472 | 0.035048 | 0.076116 | 0.11903 | 0.19964 | 0.30288 | 0.37094 | 1607.7 | 0,00 | 0,00 | 27105,00 |
| 0.075293 | 0.028785 | 0.12729 | 0.14773 | 0.073302 | 0.10323 | 0.072611 | 0.03299 | 0.10568 | 0.17602 | 0.010772 | 0.016409 | 0.049845 | 0.12783 | 0.096595 | 0.14628 | 0.14935 | 0.25445 | 0.35835 | 0.23637 | 0.38938 | 0.52154 | 923.41 | 0,00 | 33461,00 |
| 0.037457 | 0.077584 | 0.02284 | 0.10193 | 0.11385 | 0.011354 | 0.026365 | 0.048255 | 0.010732 | 0.066364 | 0.17794 | 0.078534 | 0.14424 | 0.016052 | 0.14506 | 0.07602 | 0.11645 | 0.13649 | 0.12584 | 0.13314 | 0.2357 | 0.47341 | 0.98499 | 23.677 | 26604,00 |

**5. Листинг**

// Holetsky.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

#include <iomanip>

#include <cmath>

#include <ctime>

using namespace std;

void matrixOut(float \*\* A, int n)

{

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

{

for (int j = 0; j < n + 1; j++)

cout << setprecision(5) << A[i][j] << " ";

cout << endl;

}

cout << endl;

}

int main()

{

srand(time(NULL));

int n = rand() % 6 + 20;

int m = n + 1;

float \*\*B = new float\*[n];

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

B[i] = new float[n];

float\* x = new float[n];

int s = 23;

cout << "x: " << endl;

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

{

x[i] = s + i;

cout << x[i] << " ";

}

cout << endl;

//симметричная матрица В без диагональных элементов

srand(time(NULL));

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

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

{

B[i][j] = -1 \* rand() % 5;

B[j][i] = B[i][j];

}

//диагональные элементы

float sum;

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

{

sum = 0;

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

if (i != j)

sum += B[i][j];

B[i][i] = -1\*sum + 1;

}

// A = B \* B

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

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

A[i] = new float[m];

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

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

{

float sum = 0;

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

sum += B[i][k] \* B[k][j];

A[i][j] = sum;

}

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

A[i][m - 1] = 0;

//d

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

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

A[i][m - 1] += A[i][j] \* x[j];

cout << "Input: "<< endl;

matrixOut(A, n);

//LDLT

float \* t = new float[n];

for (int k = 0; k < n - 1; k++)

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

{

t[i] = A[i][k];

float Lik = A[i][k] / A[k][k];

A[i][k] = Lik;

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

A[i][j] = A[i][j] - Lik\* t[j];

}

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

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

A[i][j] = 0;

cout << "LDLT: " << endl;

matrixOut(A, n);

//y, L \* y = d

float \* y = new float[n];

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

{

float sum = 0;

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

sum += y[j] \* A[i][j];

y[i] = A[i][m - 1] - sum;

}

cout << "y = (" << endl;

for (int k = 0; k < n; k++) {

cout << y[k];

if (k != n - 1)

cout << ", ";

}

cout << ") " << endl;

//транспонирование

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

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

{

float temp = A[j][i];

A[j][i] = A[i][j];

A[i][j] = temp;

}

float \*\* DLT = new float\*[n];

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

DLT[i] = new float[n];

//D\*LT

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

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

{

if (i == j)

DLT[i][j] = A[i][j];

else

DLT[i][j] = A[i][j] \* A[i][i];

}

//x, C \* x = y

float \* approxX = new float[n];

for (int i = n - 1; i >= 0; i--)

{

float sum = 0;

for (int j = n - 1; j > i; j--)

sum += approxX[j] \* DLT[i][j];

approxX[i] = (y[i] - sum) / DLT[i][i];

}

cout << "x\*: (" << endl;

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

{

cout << setprecision(17) << approxX[i];

if (i != n - 1)

cout << ", ";

}

cout << ") " << endl;

// Относительная погрешность решения

float approxNorm = 0;

float diffNorm = 0;

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

{

if (fabs(approxX[i]) > approxNorm)

approxNorm = fabs(approxX[i]);

if (fabs(approxX[i] - x[i]) > diffNorm)

diffNorm = fabs(approxX[i] - x[i]);

}

cout << "Relative error in norms: " << diffNorm / approxNorm << endl;

system("pause");

return 0;

}