Санкт – Петербургский политехнический университет Петра Великого

Физико-механический институт

Отчет по лабораторной работе по Численным методам
Изучение методов решения СЛАУ в MATLAB

Выполнил студент

Группы 5030301/10002 Тугай В.В.

Преподаватель: К.Н. Козлов

Санкт-Петербург

2022

Исследования, проводимые в ходе работы:

1. Разобраться в работе norm
2. Объяснить причину использования понятия обусловленности
3. Исследование матриц Гилберта различных размеров
4. Функции chol, lu, qr в MATLAB
5. Указать методы, используемые в «\»
6. Функция linsolve в MATLAB
7. Функция pcg в MATLAB
8. Использовать функцию pcg для матрицы, которая не является симметричной и положительно определенной
9. Исследование зависимости количества итерация в функции pcg от числа обусловленности матрицы, а также пример, когда количества итераций, установленных по умолчанию, не хватает для достижения заданной точности

Разобраться в работе norm

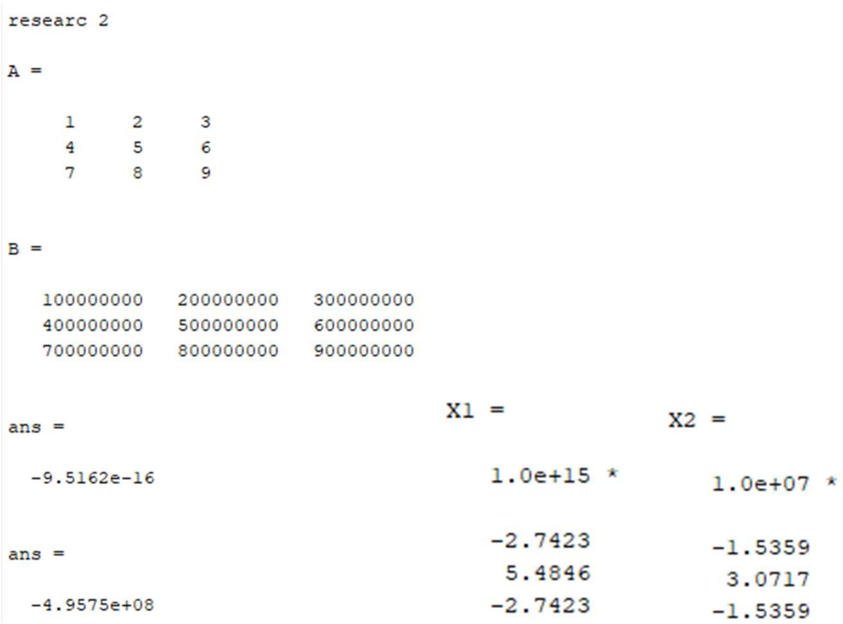
n=norm(v) – норма вектора Евклидова пространства, а n=norm(v,1) – 1-я норма. n=norm(X) – 2-я норма матрицы, n=norm(X,1) – 1-я норма матрицы, n=norm(X,”fro”) – норма Фробениуса.

```
v =

     1    -2     3

Euclidean norm of vector

n =

    3.7417

1-norm vector

n =

     6


X =

     2     0     1
    -1     1     0
    -3     3     0

2-norm of matrix

n =

    4.7234

1-norm of matrix

n =

     6

Frobenius norm

n =

     5
```

Рисунок 1. Вычисление различных видов норм

Объяснить причину использования понятия обусловленности

Число обусловленности измеряет то, как сильно значение функции может меняться при изменении аргумента, т.е. оно отражает то, как сильно функция реагирует на изменения на входе и насколько сильна ошибка на выходе связана с ошибкой на входе, а с уменьшением числа обусловленности уменьшается и неточность решения. Вырожденность матрицы же не отражает погрешность получаемого решения.



```
researc 2

A =

     1     2     3
     4     5     6
     7     8     9


B =

   100000000   200000000   300000000
   400000000   500000000   600000000
   700000000   800000000   900000000


ans =

  -9.5162e-16


ans =

  -4.9575e+08
```

```
X1 =

   1.0e+15 *

  -2.7423
   5.4846
  -2.7423
```

```
X2 =

   1.0e+07 *

  -1.5359
   3.0717
  -1.5359
```

Рисунок 2. Результат исследования №2

Исследование матриц Гилберта различных размеров

На рисунке 3 представлены отклонения получающегося решения от точного для каждого элемента матриц размерностями 5x5, 10x10, 15x15, а на рисунке 4 – невязка для каждого элемента все тех же трех матриц. На рисунке 5 можно увидеть числа обусловленности для каждой из трех матриц. С помощью невязки можно определить погрешность точного решения для СЛАУ, однако полученное решение не обязательно будут отличаться от точного на такое же число.

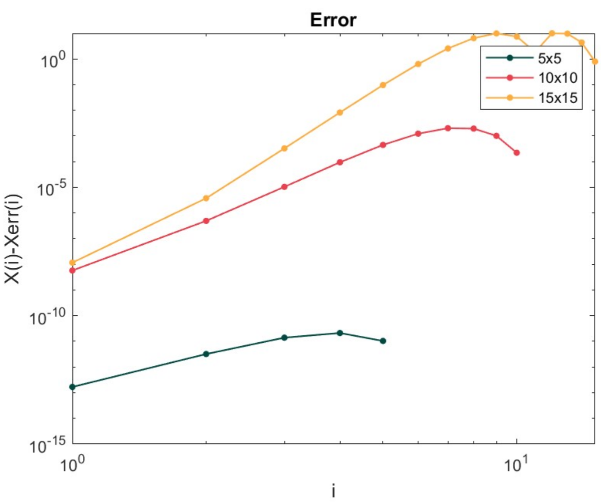


Рисунок 3. Отклонение получающегося решения от точного для всех трех матриц

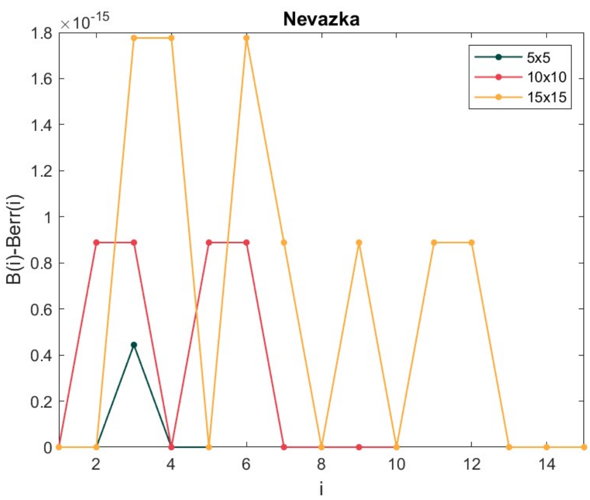


Рисунок 4. Невязка для всех трех матриц

```
cond(5x5)

ans =

   4.7661e+05

cond(10x10)

ans =

   1.6025e+13

cond(15x15)

ans =

   2.4960e+17
```

Рисунок 5. Число обусловленности для
матриц Гилберта каждого размера

Функции chol, lu, qr в MATLAB

Для функции LU нет ограничений на исходную матрицу, сама функция возвращает разложение матрицы L*U=A. Исходная матрица для функции chol должна быть симметричной и положительно-определенной, а сама функция возвращает разложение U*U T=A. Исходя из рисунка 6 функция chol реализуется быстрее, т.к. входная матрица A имеет параметры для работы функции. Максимальный размер массива в памяти составляет 1.436e+10 байт.

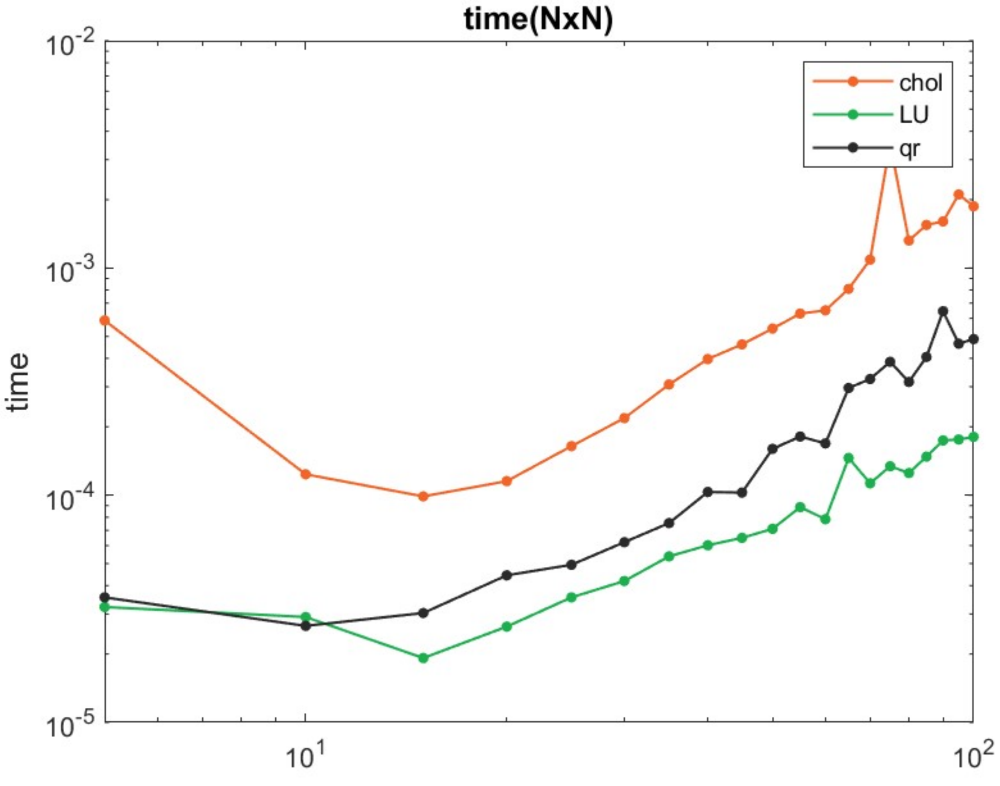


Рисунок 6. Зависимость затраченного времени
от размеров матрицы для трех функций

Указать методы, используемые в «\»

Если A-скаляр, то A\B эквивалентно A.\B. Если A-квадратная N матрица и B-матрица с N строк, то A\B решение уравнения A*x=B. Если A-матрица MxN где m~n, и B-матрица с M строк, то A\B решение методом наименьших квадратов системы уравнений A*x=B. Примеры решений можно увидеть на рисунке 7.

```
If A-scalar:

A =

     2


B =

     2
     4
     6
     8
    10


C =

     1
     2
     3
     4
     5

A\B is equivalent to A.\B
```

```
If A-square N matrix and B-matrix with N rows:

A =

     1     2     3
     4     3     1
     6     2     3


B =

     1
     1
     1


C =

         0
    0.2857
    0.1429

A\B is a solution to the equation A*x = B
```

```
If A- matrix MxN where M~N and B-matrix with M rows:

A =

     1     2     3
     4     3     1
     6     2     3
     4     5     8


B =

     1
     1
     1
     1


C =

    0.0932
    0.2328
   -0.0376

A\B is a least-squares solution to the system of equations A*x= B
```

Рисунок 7. Методы и их примеры для «\»

Функция linsolve в MATLAB

X = linsolve(A,B,opts) использует соответствующий метод, определенный структурой опций opts. Поля в opts являются логическими значениями, описывающими свойства матрицы A. Например, если A является верхней треугольной матрицей, задается opts.UT = true, чтобы заставить linsolve использовать метод, предназначенный для верхних треугольных матриц. linsolve не проверяет, обладает ли A свойствами, указанными в opts, а каждое поле может принимать значение либо true, либо false. Разумеется, не все комбинации значений полей допустимы, например матрица не может быть треугольной и в то же самое время симметричной и положительно определенной. На рисунке 8 приведен пример для двух вариантов исходной матрицы.

```
researc 6

A =

     1     0     0
     4     3     0
     6     2     3


B =

     1
     1
     1

opts.UT = true;  opts.LT = false; opts.POSDEF = false;

X =

    1.0000
    0.3333
    0.3333


A =

     1     2     3
     0     3     1
     0     0     3


B =

     1
     1
     1

opts.UT = false;  opts.LT = true; opts.POSDEF = false;

X =

    1.0000
    0.3333
    0.3333
```

Рисунок 8. Пример решения linsolve для двух вариантов исходной матрицы

Функция pcg в MATLAB

Функция X=pcg(A,B,tol,maxit,M_1,M_2,x_0) возвращает вектор-столбец решений X СЛАУ, задаваемых матрицей A и столбцом свободных членов B. tol – допустимая погрешность. maxit - максимальное количество итераций. M1 и M2 -матрицы предварительной подготовки, которые могут обеспечить буструю сходимость функции pcg. x_0 - начальное приближение. Также функцию можно задать так: [x, flag, relres, iter, resvec] = pcg(A,B,tol,maxit,M_1,M_2,x_0), где в квадратных скобках выходные данные. glag - флаг сходимости, который указывает, было ли получено решение. relres - относительная остаточная ошибка norm(B-A*X)/norm(B). iter – номер итерации, на которой pcg прекратил свою работу. resvec – остаточная ошибка norm(B-A*X).

Использовать функцию pcg для матрицы, которая не является симметричной и положительно определенной

Если матрица на входе не является положительно определенной и симметричной, то flag=4, пример такой ситуации можно увидеть на рисунке 9. В таком случае pcg возвращает решение X с минимальной нормой невязки, вычисленной по всем доступным итерациям.

```
researc 8

A =

    0.8003    0.9595    0.6787    0.1712    0.0971
    0.1419    0.6557    0.7577    0.7060    0.8235
    0.4218    0.0357    0.7431    0.0318    0.6948
    0.9157    0.8491    0.3922    0.2769    0.3171
    0.7922    0.9340    0.6555    0.0462    0.9502


X =

     1
     1
     1
     1
     1


B =

    2.7068
    3.0849
    1.9273
    2.7511
    3.3781


tol =

   1.0000e-16


maxit =

   100
```

```
Flag_k =

     4


Error =

    0.0127

solutions from pcg
    0.9882
    1.0564
    0.8476
    0.9962
    1.1092
```

Рисунок 9. pcg для матрицы, которая не является
симметричной и положительно определенной

Исследование зависимости количества итерация в функции pcg от числа обусловленности матрицы, а также пример, когда количества итераций, установленных по умолчанию, не хватает для достижения заданной точности

flag=1 свидетельствует о том, что количества итераций не хватило для достижения заданной точности, пример такой ситуации можно увидеть на рисунке 10. Также, исходя из рисунка 11, можно заметить, что график зависимости количества итераций от числа обусловленности ведет себя скачкообразно, однако решение все равно находится для числа обусловленности от 1 до 51 с шагом в 10.

```
%9
disp('researc 9')

A=rand(10);
[u,d,v]=svd(A);
d=eye(10);
d(1,1)=10;
A=u*d*v;
A=A*A';
X=ones(10,1);
B=A*X;
tol=0.0000000000000001
[x,flag,relres,iter] = pcg(A,B,tol,2);
disp('relres, iter, flag')
disp(relres)
disp(iter)
disp(flag)


relres, iter, flag
   3.2979e-15

     2

     1
```

Рисунок 10. Код и результат выполнения кода для случая нехватки итераций

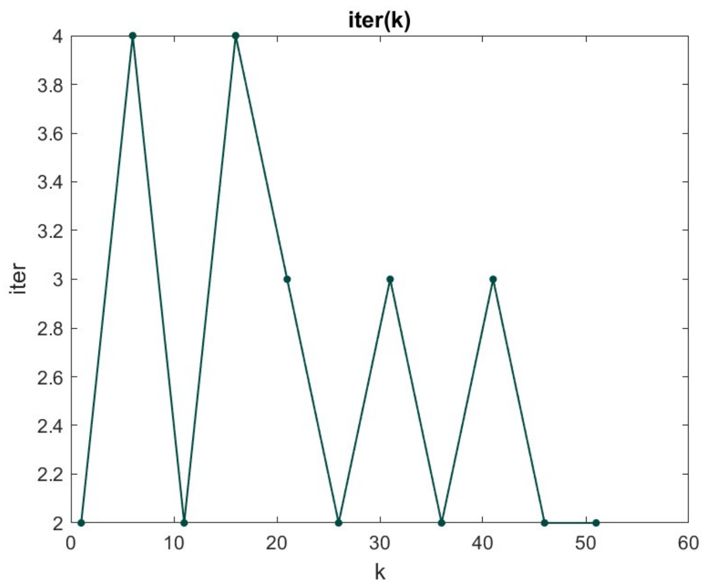


Рисунок 11. Зависимость количества итераций от числа обусловленности

Вывод

В ходе проделанной лабораторной работы нами были исследованы методы решения СЛАУ различными функциями из пакета MATLAB. По результатам, полученным в ходе работы данных функций, были построены графики различный зависимостей. В итоге можно прийти к выводу, что использованные нами функции показали высокую точность в решении СЛАУ.