Лабораторная работа 12
Решение задач линейной алгебры

Решение систем линейных алгебраических уравнений (СЛАУ)

Команда \ или mldivide
Обратное деление:
X = A\B (\ - backslash) - является решением матричного уравнения AX = B
Функция mldivide равносильна операции обратного деления (\)

Прямое деление: X = B/A - является решением матричного уравнения XA = B.

СЛАУ Ax=b, где A – квадратная матрица размера nxn, b – вектор правой части размера nx1, x – вектор неизвестных размера nx1 может:

1) иметь единственное решение,
2) иметь бесконечно много решений,
3) не иметь решений.

Систему называют определенной, если она имеет единственное решение.

Проверка точности решения: вычисление *невязки* и нормы вектора невязки. Если x является приближенным решением уравнения A*x=b, то разница между левой и правой частью будет почти нулевой, т.е. вектор невязки r=b-A*x будет близок к нулевому вектору, а норма вектора невязки близка к нулю. Для точного решения невязка будет нулевой в отсутствие ошибок округления.

Пример 1. Решение СЛАУ методом Гаусса

```
A = [4 1 2; 3 7 1; 2 2 8];
b = [7; 11; 12];
x1 = A\b % в Matlab для вычисления x по умолчанию реализован
алгоритм Гаусса
% функция mldivide равносильна операции обратного деления (\)
x2 = mldivide(A, b)

% вычислим вектор невязки и норму вектора невязки
r1=b - A*x1, norm(r1) %norm(v) – Евклидова норма, если v – вектор,
эквивалентна команде norm(v,2); см. help
r2=b - A*x2, norm(r2)

%Аналитическое решение, символьные вычисления
As = sym(A) % преобразуем матрицу в символьную
bs=sym(b)
xs=As\bs

% вычислим вектор невязки и норму вектора невязки
rs=bs - As*xs, norm(rs)

%Выполним то же для произвольной матрицы
A=rand(4)
b=rand(4,1)
x=A\b
r=b-A*x
norm(r)

%Сравним точное и приближенное решение в длинном формате
format long
```

```
A=rand(3)
x_ex=rand(3,1)% зададим точное решение
b=A*x_ex %вычислим правую часть
x=A\b % приближенное решение
x-x_ex % разница между точным и приближенным решением (ошибка)
norm(x-x_ex)
```

Пример 2. Обращение матриц
Если матрица A является квадратной и невырожденной (определитель отличен от нуля), уравнения AX =E и XA =E (E – единичная матрица) имеют одинаковое решение X . Это решение называется матрицей, обратной к матрице A , обозначается через A^{-1} и вычисляется с помощью функции `inv(A)` или `A^(-1)`

Матрица называется вырожденной, если ее столбцы или строки линейно зависимы, определитель такой матрицы равен нулю. Для вырожденной матрицы не существует обратной.

В Matlab при попытке вычислить обратную матрицу для вырожденной матрицы (по-английски singular) или матрицы, близкой к вырожденной, выдается предупреждение:
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

```
A= randi(10,4,4)
d = det (A) % вычисляется определитель A
X = inv (A) % вычисляется обратная матрица к A
E=eye(size(A)) %конструируем единичную матрицу
A*X== E % объясните результат
format long
A*X-E % посмотрите разницу между правой и левой частью
norm(A*X-E) % чему равна норма?
format short

%Вычисление обратной матрицы методом обратного деления (как решение
уравнения AX=E)
X1=A\E
norm(A*X1-E)

%Пример вырожденной матрицы
A = rand(4);
A(:,2) = 2*A(:,1) %столбцы 1 и 2 линейно зависимы
d = det (A) % определитель близок к нулю
X = inv (A) % что выдается?
% Посмотрите разницу между правой и левой частью и проанализируйте
результат
A*X-E
norm(A*X-E)
```


Нахождение собственных значений и собственных векторов в аналитическом и численном виде.

Пример 3. Спектральный анализ матрицы A
Собственным значением и собственным вектором квадратной матрицы A называются скаляр λ и вектор v, удовлетворяющие условию Av = λ v. Для вычисления собственных значений и векторов используется команда **eig**. Совокупность всех собственных значений называется спектром матрицы. У вещественной матрицы

размера n будет n собственных значений. Среди них могут быть кратные и комплексные.

charpoly(A,x) – характеристический многочлен матрицы A относительно переменной x, работает как для символьной, так и для численной матрицы

- характеристический многочлен $P_A(\lambda) = \det(\lambda E - A)$

Корни характеристического уравнения являются собственными значениями матрицы. Для нахождения корней, т.е. решения уравнения $P_A(\lambda) = 0$ применяются команды **solve** (аналитическое решение) и **vpasolve** (численное решение)

```
A = [4 1 2; 3 7 1; 2 2 8];
lambda = eig(A) % здесь собственные значения различны и вещественны
% и выходной параметр - вектор
[V,D] = eig(A) % собственные значения на главной диагонали
диагональной матрицы D, и все собственные векторы в матрице V
записаны по столбцам, они линейно независимы

%аналитическое вычисление собственных значений
A1 = sym(A) % преобразуем матрицу в символьную
Lambda1 = eig(A1)
% численные значения собственных векторов
% сравните результат двух команд
double(lambda1)
vpa(lambda1)

%характеристический многочлен
charpoly(A) % A - численная, выдает коэффициенты многочлена
syms x
polyA=charpoly(A,x) % выдает многочлен
eigenA = solve(polyA) % аналитическое нахождение корней хар.
многочлена
double(eigenA)
eigenA = vpasolve(polyA) % численное решение характеристического
уравнения

charpoly(A1) % A - символьная
syms x
polyA1=charpoly(A1,x) % выдает многочлен
```


Пример 4. Факторизация матриц

```
A = [4 1 2; 3 7 1; 2 2 8];
b = [7; 11; 12];
[l,u]=lu(A) % lu-факторизация A на произведение l – нижней
треугольной и u – верхней треугольной матриц, такая что l*u=A
[Q,R]=qr(A) % qr-факторизация A; R – верхняя треугольная, Q –
унитарная или в вещественном случае ортогональная матрица(Q*Q'=E, E
– единичная матрица), такая, что Q*R=A
B=A*A'  % B – симметричная матрица
issymmetric(B)
R=chol(B) % если B – положительно определенная матрица, то R'*R = B,
где R – верхне-треугольная матрица

%Проверим ортогональность матрицы Q, т.е. что Q^(-1)=Q' или что
Q^(-1)*Q=E
```

```
isequal(Q^(-1),Q')% проверка равенства двух матриц, что выдает:
true или false?
format long %сравните результаты в длинном формате
Q^(-1)==Q'
Q^(-1)-Q'% разность
norm(Q^(-1)-Q') % норма разности
%Проверим, что Q^(-1)*Q=E
Q^(-1)*Q % сравните с единичной матрицей
E=eye(3)
isequal(Q^(-1)*Q,E)% почему не true?
Q^(-1)%Q-E% разность
norm(Q^(-1)%Q-E) % норма разности
```

Пример 5. Решение СЛАУ с помощью процедуры linsolve

```
clc
clear
A = [4 1 2; 3 7 1; 2 2 8];
b = [7; 11; 12];
x=linsolve(A,b) % изучите справку linsolve – как управлять выбором
опций для этой команды в зависимости от свойств матрицы системы
```

Задание 1.

Записать три СЛАУ в матричном виде. Вычислить определитель и ранг (команда rank) для каждой матрицы. Для каждой системы найти решение численно и аналитически (символьные вычисления). Для обоих решений вычислить невязку и норму невязки. Для каждой системы написать в комментариях, какому случаю существования решения она соответствует.

1) $\begin{cases} 3x_1 - 2x_2 - 5x_3 + x_4 = 3, \\ 2x_1 - 3x_2 + x_3 + 5x_4 = -3, \\ x_1 + 2x_2 - 4x_4 = -3, \\ x_1 - x_2 - 4x_3 + 9x_4 = 22. \end{cases}$ здесь $A = \begin{bmatrix} 3 & -2 & -5 & 1 \\ 2 & -3 & -1 & 5 \\ 1 & 2 & 0 & -4 \\ 1 & -1 & -4 & 9 \end{bmatrix}$

2) $\begin{cases} 4x_1 - 3x_2 + 2x_3 - x_4 = 8, \\ 3x_1 - 2x_2 + x_3 - 3x_4 = 7, \\ 2x_1 - x_2 - x_4 = 6, \\ 5x_1 - 3x_2 + x_3 - 8x_4 = 1. \end{cases}$

3) $\begin{cases} 2x_1 + 3x_2 - x_3 + x_4 = 1, \\ 8x_1 + 12x_2 - 9x_3 + 8x_4 = 3, \\ 4x_1 + 6x_2 + 3x_3 - 2x_4 = 3, \\ 2x_1 + 3x_2 + 9x_3 - 7x_4 = 3. \end{cases}$

Задание 2.

Для трех матриц из предыдущего задания вычислить обратные матрицы тремя методами: `A^(-1)`, `inv(A)` и `A\eye(n)`, где n – размер матрицы. Сравнить результаты.

Задание 3.

Дана матрица $A=\begin{bmatrix}5&4&3&2&1\\4&8&6&4&2\\3&6&9&6&3\\2&4&6&8&4\\1&2&3&4&5\end{bmatrix}$. Найти ее собственные значения аналитически и численно. Найти характеристический многочлен и его корни. Сравнить все результаты.

Задание 4.

Задана случайная вещественная матрица четвертого порядка. Найти ее собственные значения и собственные векторы. Проверить, что каждый собственный вектор u, соответствующий собственному числу λ удовлетворяет уравнению $Au=\lambda u$ (подставить собственное значение и соответствующий ему собственный вектор в уравнение и вычислить невязку и норму невязки).

Задание 5.

Задана случайная вещественная матрица A четвертого порядка. Найти ее LU-факторизацию. Проверить, что A=LU. Найти ее QR-факторизацию. Проверить, что A=QR. Проверить, что матрица Q ортогональна.

Задание 6.

Рассмотрите СЛАУ в матричной форме Ax=b, где A – случайная матрица размера n×n (ф-я rand), b – случайный вектор. Сначала можно рассмотреть небольшой порядок системы (n=4). Затем исследуйте случаи, когда порядок системы n=1000, 2000, 3000 и выше. Найдите решение системы с помощью способов, описанных ниже. Для каждого способа нахождения решения вычислите норму невязки и время получения решения (команды tic, toc). Сформулируйте выводы по полученным результатам в комментариях к скрипту (какой метод самый быстрый, самый медленный, какой метод дает наилучшую точность).

1) Решение с помощью команды mldivide или \
2) Решение с помощью команды linsolve
3) Решение с помощью нахождения обратной матрицы
4) Решение с помощью LU-факторизации (A=LU): факторизуйте матрицу A с помощью процедуры [L,U]=lu(A) и на основе факторизованной системы L*U*x=b получите решение. При измерении времени решения рассмотрите случаи, когда время нахождения факторизации учитывается и не учитывается:

```
%A=LU; Ax=b => x=U^(-1)*L^(-1)*b
%tic %если замерять время получения факторизации
[L,U]=lu(A);
tic %если не замерять время получения факторизации
x4=U\(L\b);
toc
```

5) Решение с помощью QR-факторизации (A=QR): факторизуйте матрицу A с помощью процедуры [Q,R]=qr(A) и на основе факторизованной системы Q*R*x=b получите решение. При измерении времени решения рассмотрите случаи, когда время нахождения факторизации учитывается и не учитывается:

```
%A=QR; x=R^(-1)*Q^(-1)*b или x=R^(-1)*Q'*b, т.к. Q^(-1)=Q'
%tic %если замерять время получения факторизации
```

```
[Q,R]=qr(A);
tic %если не замерять время получения факторизации
x5=R\(Q'*b); % первый способ
%x55=R\(Q\b); % или второй способ
toc
norm(b-A*x5)
% что вычисляется быстрее, x5 или x55?
```

Задание на дополнительные баллы (1 балл)

Задание 7.

Рассмотрите СЛАУ в матричной форме Ax=b, где A – случайная симметричная положительно определенная матрица размера n×n (ф-я rand), b – случайный вектор. Сначала можно рассмотреть небольшой порядок системы (n=4). Затем исследуйте случаи, когда порядок системы n=1000, 2000, 3000 и выше. Найдите решение системы с помощью способов, описанных ниже. Для каждого способа нахождения решения вычислите норму невязки и время получения решения (команды tic, toc). Сформулируйте выводы по полученным результатам в комментариях к скрипту (какой метод самый быстрый, самый медленный, какой метод дает наилучшую точность).

1) Решение с помощью команды mldivide или \
2) Решение с помощью команды linsolve (использовать linsolve(A,b,opts) с опцией положительной определённости opts.SYM=true, opts.POSDEF=true)\
3) Решение с помощью разложения Холецкого (A=LL’ для факторизации с нижне-треугольной матрицей или A=RR’ для факторизации с верхне-треугольной матрицей): факторизуйте матрицу A с помощью процедуры R=chol(A) и на основе факторизованной системы $L*L^T*x=b$ или $R^T*R*x=b$ получите решение. При измерении времени решения рассмотрите случаи, когда время нахождения факторизации учитывается и не учитывается:

```
% A=R'*R; x=R^(-1)*R'^(-1)*b
%tic %если замерять время получения факторизации
[R,p]=chol(A); % что показывает p? Изучите справку
tic %если не замерять время получения факторизации
x2=R\(R'\b);
toc
```