

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана»
(национальный исследовательский университет)

Кафедра «Прикладная математика»



Домашняя работа

по дисциплине «Методы численного решения задач
линейной алгебры»

Вариант 7

Выполнила студентка группы ФН2-31М

Матюхина Д.И.

Преподаватель

Родин А.С.

1. Постановка задачи

Нужно сформировать матрицу размером 10×10 по следующему принципу. В качестве базовой матрицы A_0 берется известная матрица, которая получается после дискретизации одномерного оператора Лапласа методом конечных разностей или методом конечных элементов на равномерной сетке:

$$A_0 = \begin{pmatrix} 2 & -1 & 0 & 0 & & & & & & \\ -1 & 2 & -1 & 0 & & & & & & \\ 0 & -1 & 2 & -1 & & & & & & \\ & & & & \ddots & & & & & \\ & & & & & -1 & 2 & -1 & 0 & \\ & & & & & 0 & -1 & 2 & -1 & \\ & & & & & 0 & 0 & -1 & 2 & \end{pmatrix}.$$

Для данной матрицы известны аналитические формулы для собственных значений ($n = 10$)

$$\lambda_j^0 = 2 \left(1 - \cos \frac{\pi j}{n+1} \right), \quad j = 1, \dots, n,$$

и компонент собственных векторов (вектора имеют 2-норму равную 1)

$$z_j^0(k) = \sqrt{\frac{2}{n+1}} \sin \frac{\pi j k}{n+1}, \quad k = 1, \dots, n.$$

Итоговая матрица получается по формулам:

$$\begin{aligned} A &= A_0 + \delta A, \\ \delta A &= \begin{cases} \frac{c}{i+j}, & i \neq j \\ 0, & i = j \end{cases}, \\ c &= \frac{N_{var}}{N_{var} + 1} \varepsilon, \end{aligned}$$

где $N_{var} = 7$ – номер варианта, ε – параметр, значение которого задается далее.

Нужно выполнить следующие задания:

1. Взять матрицу A для значения $\varepsilon = 0.1$, убрать последний столбец и сформировать из первых 9 столбцов матрицу \hat{A} размера 10×9 . Решить линейную задачу наименьших квадратов для вектора невязки

$$r = \hat{A}x - b,$$

где вектор b размерности 10×1 нужно получить по следующему алгоритму: выбрать вектор x_0 размерности 9×1 и для него вычислить $b = \hat{A}x_0$.

Для решения поставленной задачи использовать алгоритм QR-разложения, основанный на методе отражений Хаусхолдера.

После получения решения сделать оценку величины $\|x - x_0\|_2 / \|x_0\|_2$.

2. Для матрицы A найти все ее собственные значения ($\lambda_j, j = 1, \dots, 10$) и собственные вектора ($z_j, j = 1, \dots, 10$ с 2-нормой равной 1) с помощью неявного QR-алгоритма со сдвигом (с предварительным приведением матрицы к форме Хессенберга) для трех вариантов: $\varepsilon = 10^{-1}, 10^{-3}, 10^{-6}$.

По итогам расчетов нужно сделать сводную таблицу, в которой указать следующие величины: $|\lambda_j - \lambda_j^0|$ и $\|z_j - z_j^0\|_2$ для $j = 1, \dots, 10$.

2. Задание 1

Линейная задача наименьших квадратов (для матрицы A размером $m \times n$ и m -вектора b) заключается в отыскании n -вектора x , минимизирующего величину $\|Ax - b\|_2$. Один из способов решения задачи (для случая $m > n$) – использование QR -разложения матрицы A .

Теорема. Пусть A – матрицы размера $m \times n$, причем $m \geq n$. Предположим, что A имеет полный столбцовый ранг. Тогда существуют и единственны $m \times n$ -матрица Q с ортонормированными столбцами и верхнетреугольная $n \times n$ -матрица R с положительными диагональными элементами r_{ii} , такие, что $A = QR$.

Формула для вектора x , минимизирующего функцию $\|r\|_2 = \|Ax - b\|_2$, может быть получена из формулы для решения системы нормальных уравнений

$$x = (A^T A)^{-1} A^T b = (R^T Q^T Q R)^{-1} R^T Q^T b = (R^T R)^{-1} R^T Q^T b = R^{-1} (R^T)^{-1} R^T Q^T b = R^{-1} Q^T b.$$

Метод отражений Хаусхолдера заключается в следующем. Матрица отражений $P = I - 2uu^T$ ($\|u\|_2 = 1$) ортогональна, симметрична и любой ненулевой вектор x преобразует в вектор, пропорциональный первому вектору естественного базиса

$$Px = ce_1.$$

Вектор Px является отражением вектора x относительно плоскости, проходящей через 0 перпендикулярно к u .

Поэтому мы будем последовательно действовать на исходную матрицу A матрицей отражений P_i и обнулять элементы, находящиеся ниже главной диагонали в соответствующих столбцах $A^{(i)}, i = 1, \dots, n$. При этом при каждом последующем отражении мы не меняем уже полученных нулей в предыдущих столбцах j ($j < i$). В итоге будет получена верхнетреугольная $m \times n$ матрица \tilde{R} и ортогональная $m \times n$ матрица Q , которая образована первыми n столбцами матрицы $\prod_{i=1}^n P_i$ (P_i имеют размерность $m \times m$). Для последующего решения системы $Rx = Q^T b$ мы отбросим нулевые строки в матрице \tilde{R} .

Код программы приведен ниже.

```
n = 10;
```

```
%Формируем базовую матрицу
```

```
A = zeros(n);
```

```
for i=1:n
```

```
    for j = 1:n
```

```
        if i == j
```

```
            A(i,j) = 2;
```

```
        elseif (j == i + 1) || (j == i - 1)
```

```
            A(i,j) = -1;
```

```
        end
```

```
    end
```

```
end
```

```

%Формируем возмущенную матрицу
eps = 1e-1;
c = 7*eps/8;

for i=1:n
    for j = 1:n
        if i ~= j
            A(i,j) = A(i,j) + c/(i+j);
        end
    end
end

%Задание 1
A_rect = A(:, 1:n-1); %прямоугольная матрица
x0 = rand(n-1,1);
b = A_rect*x0;

[Q1, R1, b1] = Householder(A_rect, b);

x = reverse_Gauss(R1, b1);

prec = norm(x-x0)/norm(x0); %относительная погрешность

function [Q, R, b] = Householder(A, b)
[m,n] = size(A);
for i = 1:n
    u = House(A(i:m, i));
    Pi = eye(m+1-i) - 2*u*(u');
    P = eye(m);
    if i == 1
        Q = P;
    end
    A(i:m, i:n) = Pi*A(i:m, i:n);
    gamma = -2*(u')*b(i:m);
    b(i:m) = b(i:m) + gamma*u;
    P(i:m, i:m) = Pi;
    Q = Q*P;
end
R = A(1:n,:);
b = b(1:n);
Q = Q(:,1:n);
end

function [u] = House(y)
u = y;
u(1) = y(1) + sign(y(1))*norm(y);
u = u / norm(u);

```

```

end

%Обратный ход метода Гаусса
function [y] = reverse_Gauss(A,b)
%Ay = b
n = size(b,1);
y = zeros(n,1);
for i = n:-1:1
    y(i) = (b(i) - sum(A(i,i+1:n)*y(i+1:n)))/A(i,i);
end
end

```

По окончании работы программы получаем следующие результаты:

prec	2.5247e-15
Q1	10x9 double
R1	9x9 double
x	[0.8147;0.9058;0.1270;0.9134;0.6324;0.0975;0.2785;0.5469;0.9575]
x0	[0.8147;0.9058;0.1270;0.9134;0.6324;0.0975;0.2785;0.5469;0.9575]

Рис. 1. Найденное решение задачи x и случайно сформированный начальный вектор x_0

3. Задание 2

QR-итерация со сдвигом заключается в следующем. Для матрицы A последовательно выполняются действия:

1. выбирается сдвиг σ_i ;
2. выполняется разложение $A_i - \sigma_i I = Q_i R_i$;
3. $A_{i+1} = R_i Q_i + \sigma_i I$.

Для поиска собственных значений будем использовать QR-итерацию с неявными сдвигами. Матрицу A сначала приводим к верхней форме Хессенберга H методом отражений Хаусхолдера, то есть обнуляем элементы, находящиеся ниже первой поддиагонали. В нашем случае матрица A симметричная и в процессе приведения к верхней форме Хессенберга станет трехдиагональной. Алгоритм будет неявным, поскольку QR-разложение матрицы A не будет вычисляться в явном виде.

Матрицы A_i и A_{i+1} ортогонально подобны и их собственные значения совпадают. Алгоритм определяется следующей формулой

$$A_{i+1} = R_i Q_i + \sigma_i I = Q_i^T Q_i R_i Q_i + \sigma_i Q_i^T Q_i = Q_i^T (Q_i R_i + \sigma_i I) Q_i = Q_i^T A_i Q_i.$$

Алгоритм считается сошедшимся, если элементы $A_{i+1}(n, 1 : n - 1)$ достаточно малы (мы будем проверять малость элемента $A_{i+1}(n, n - 1)$). При этом в позиции $A_{i+1}(n, n)$ находится собственное значение λ_i . Затем проводится редукция матрицы от размера $n \times n$ к размеру $(n - 1) \times (n - 1)$ и определяется следующее собственное значение по тому же алгоритму.

Собственные вектора ищем методом обратной итерации (обратным степенным методом)

$$y_{i+1} = (A - \sigma_i I)^{-1} x_i;$$

$$x_{i+1} = y_{i+1} / \|y_{i+1}\|.$$

Выбирая сдвиг σ_i вблизи уже найденного собственного значения λ_i , получаем собственный вектор x_i , ассоциированный с собственным значением λ_i .

Стоит отметить, что для вычисления y_{i+1} мы будем использовать обратный ход метода Гаусса, поскольку

$$(A_i - \sigma_i I)y_{i+1} = x_i;$$

$$QRy_{i+1} = x_i;$$

$$Ry_{i+1} = Q^T x_i.$$

Код программы приведен ниже.

```
%Задание 2
%Собственные значения
H = Hessenberg(A);
eigval = zeros(n,1);
iter = [];

for i = n:-1:2
    [H, iter1, eigval(i)] = implicit_QR(H(1:i,1:i));
    iter(i) = iter1;
end
eigval(1) = H(1,1);
eigval = sort(eigval);

j = 1:n;
lambda = 2*(1-cos(pi*j/(n+1)))'; %аналитические формулы
prec1 = abs(lambda-eigval);

%Собственные вектора
eigvect = eigVec(A, eigval);
z=zeros(n,n);
for k=1:n
    z(k,:)=(sqrt(2/(n+1))*sin((pi.*j*k)/(n+1))); %аналитические формулы
end

prec2 = zeros(n,1);
for i = 1:n
    prec2(i) = norm(z(:,i)-eigvect(:,i));
end

function [A] = Hessenberg(A)
n = size(A,1);
for i = 1:n-2
```

```

    u = House(A(i+1:n, i));
    Pi = eye(n-i) - 2*u*(u');
    A(i+1:n, i:n) = Pi*A(i+1:n, i:n); %не меняется 1 строка
    A(1:n, i + 1:n) = A(1:n, i+1:n)*Pi; %не меняется 1 столбец
end
end

function [A, iter, eigval] = implicit_QR(A)
n = size(A,1);
iter = 0;
while abs(A(n,n-1)) > 1e-10
    sigma = A(n,n);
    A = A - sigma*eye(n);
    for i=1:n-1
        Q = eye(n);
        c = A(i,i)/norm([A(i,i), A(i+1,i)]);
        s = A(i+1,i)/norm([A(i,i), A(i+1,i)]);
        Q(i:i+1, i:i+1) = [c -s; s c];
        %QT = Q'; %Вместо A = Q'AQ
        A = (Q')*A*Q;
        %A(i:i+1, :) = QT(i:i+1,:)*A(:,:); %Вместо A = Q'AQ
        %A(:, i:i+1) = A(:,:)*Q(:,i:i+1); %Вместо A = Q'AQ
    end
    A = A + sigma*eye(n);
    iter = iter + 1;
end
eigval = A(n,n);
end

function [X] = eigVec(A, lambda)
n = size(lambda,1);
X=zeros(n,n);
lambda = lambda - 0.01;
for i = 1:n
    x = zeros(n,1);
    y = x;
    y(1) = 1;
    [Q, R] = Householder(A - lambda(i)*eye(n), y);
    while(norm(x-y)>1e-10)
        x=y;
        y = reverse_Gauss(R,Q'*x);
        y=y/norm(y);
    end
    X(:,i)=x;
end
end

```

Таблица 1. Результаты расчетов при $\varepsilon = 10^{-1}$

λ	$ \lambda_j - \lambda_j^0 $	$\ z_j - z_j^0\ _2$
0.1463	0.0653	0.1192
0.3212	0.0037	0.1087
0.6945	0.0043	0.0594
1.1614	0.0077	0.0141
1.7072	0.0082	0.0153
2.2728	0.0118	0.0136
2.8189	0.0119	0.0170
3.2971	0.0126	0.0203
3.6709	0.016	0.0238
3.9095	0.0095	0.0183

Таблица 2. Результаты расчетов при $\varepsilon = 10^{-3}$

λ	$ \lambda_j - \lambda_j^0 $	$\ z_j - z_j^0\ _2$
0.0817	6.9201e-04	9.5212e-04
0.3175	2.0356e-05	8.3362e-04
0.6903	2.5625e-05	5.2720e-04
1.1691	7.8816e-05	1.2936e-04
1.7153	8.4100e-05	1.4738e-04
2.2845	1.1794e-04	1.3566e-04
2.8307	1.1948e-04	1.7015e-04
3.3096	1.2575e-04	2.0389e-04
3.6824	1.1648e-04	2.4046e-04
3.9189	9.5425e-05	1.8599e-04

Таблица 3. Результаты расчетов при $\varepsilon = 10^{-6}$

λ	$ \lambda_j - \lambda_j^0 $	$\ z_j - z_j^0\ _2$
0.0810	6.9234e-07	9.5013e-07
0.3175	2.0225e-08	8.3157e-07
0.6903	2.5472e-08	5.2657e-07
1.1692	7.8830e-08	1.2929e-07
1.7154	8.4119e-08	1.4733e-07
2.2846	1.1794e-07	1.3566e-07
2.8308	1.1948e-07	1.7016e-07
3.3097	1.2575e-07	2.0390e-07
3.6825	1.1649e-07	2.4057e-07
3.9190	9.5434e-08	1.8599e-07

Количество итераций до редукции матрицы

$$iter1 = [3, 5, 5, 4, 2, 2, 2, 2, 2]; \quad iter2 = [8, 2, 4, 5, 4, 4, 2, 2, 2];$$

$$iter3 = [15, 1, 3, 3, 3, 4, 4, 3, 2].$$

























	A	10x10 double
	A_rect	10x9 double
	b	[1.7722;-1.6203;0.8264;0.4584;-0.7867;0.9733;-0.9383;-0.2141;1.4097;-0.9157]
	b1	[-2.3097;1.0204;0.1158;-0.9311;0.8490;-1.0511;-0.0913;1.3244;-1.0643]
	c	8.7500e-07
	eigval	[0.0810;0.3175;0.6903;1.1692;1.7154;2.2846;2.8308;3.3097;3.6825;3.9190]
	eigvect	10x10 double
	eps	1.0000e-06
	H	[0.0810,4.6206e-17;-1.6926e-19,0.3175]
	i	10
	iter	[0,2,3,4,4,3,3,1,15]
	iter1	2
	j	[1,2,3,4,5,6,7,8,9,10]
	k	10
	lambda	[0.0810;0.3175;0.6903;1.1692;1.7154;2.2846;2.8308;3.3097;3.6825;3.9190]
	n	10
	prec	1.9698e-15
	prec1	[6.9234e-07;2.0225e-08;2.5472e-08;7.8830e-08;8.4119e-08;1.1794e-07;1.1948e-07;1.2575e-07;1.1649e-07;9.5434e-08]
	prec2	[9.5013e-07;8.3157e-07;5.2657e-07;1.2929e-07;1.4733e-07;1.3566e-07;1.7016e-07;2.0390e-07;2.4057e-07;1.8599e-07]
	Q1	10x9 double
	R1	9x9 double
	x	[0.9649;0.1576;0.9706;0.9572;0.4854;0.8003;0.1419;0.4218;0.9157]
	x0	[0.9649;0.1576;0.9706;0.9572;0.4854;0.8003;0.1419;0.4218;0.9157]
	z	10x10 double

Рис. 2. Результаты работы программы при $\varepsilon = 1e - 6$