Операции с матрицами, свойства матриц, решение СЛАУ

В конце файла со всеми заданиями для пяти команд, вы обнаружите Приложения (целевой Help). Стоит в него заглянуть!

Команда 1.

1) Загрузите двоичный файл с данными V1.mat.
   a) Определите структуру матриц A и B визуально, их свойства (определитель, симметрию, существуют ли предпосылки переходить для вычислений в класс разреженных матриц).
   b) Блок матрицы, состоящий только из побочной диагонали, отредактируйте так, чтобы его диагональ стала главной и поместите её на своё место в блочно-диагональной матрице. Вычислите определитель, оставив идентификатор исходной матрицы.
   c) Сгенерируйте (или задайте) вектор решения (одинаковый для обеих матриц A и B), вычислите произведение матриц на столбец решения, получите столбцы свободных членов a и b для двух линейных систем.
2) Решите обе системы встроенным методом Гаусса. Определите абсолютную и относительную точности решений в обоих случаях.
3) В случае симметричной матрицы используйте для решения факторизованную систему; см. Приложение 1.
4) Выведите решение (кажется ;), оба должны совпадать?) в текстовый файл в формате с плавающей запятой, каждый элемент решения в новой строке пронумерован, например,

   1) 2.e-1
   2) 3.14e0

   ….

Задание для конструирования пользователем системы, а также её решения с помощью факторизации матрицы и решателя linsolve.

1) Конструирование системы с положительно определенной матрицей.
   a) *выберите высокий порядок (n>1000) матрицы* `A` *и используйте для её построения функцию* `rand` *или* `randi` *с положительными элементами.*
   `b)` *На основе вашей матрицы* A, *сконструируйте положительно определенную матрицу* P, используя критерий, указанный в замечании Приложения 1. *Покажите, что P является положительно определенной.*

Для матрицы P вычислите вектор-столбец `b` системы `Px=b` (*столбец свободных членов вы уже умеете строить!*). Решите систему с опцией, предусматривающей положительную определённость матрицы системы

```
x=linsolve(P,b,opts)
```

Оцените время решения и сравните с встроенным в ML методом Гаусса.

Замечание 1. Положительно-определенная матрица получится, если она ***симметричная****,* ***имеет положительные элементы****, а также* ***диагональное преобладание*** *(т.е. диагональные элементы много больше суммы всех остальных элементов в строке).*

Замечание 2. Пусть tG – время решения системы методом Гаусса, tPosDef – с факторизацией по Холецкому с помощью `linsolve`*.*

Время решений для сравнения – строка ML:

['tG=', num2str(tG), ' tPosDef= ', num2str(tPosDef)]

Функции определяющие эффективность (скорость) работы алгоритма вы найдёте в Приложении 3.

В конце файла со всеми заданиями для пяти команд, вы обнаружите Приложения (целевой Help). Стоит в него заглянуть!

Команда 2.

1) Загрузите двоичный файл с данными V2.mat.Определите элементы Workspace, их свойства (визуально – структуру матриц, определитель, симметрию, процент ненулевых элементов матриц, объясните, зачем это может быть интересно).
   b) Выберите вырожденную матрицу. Выясните, какой блок матрицы является причиной её вырождения. Отредактируйте блок так, чтобы его элементы расположились вдоль главной диагонали блочной матрицы.
   c) Задайте любым способом вектор решения (одинаковый для обеих матриц A и B), вычислите произведение матриц на вектор решения, получите их столбцы свободных членов a и b.
2) Решите систему, которая является симметричной, методом qr-факторизации, а несимметричную – методом Гаусса (без lu-факторизации); см. Приложение 1. Определите абсолютную и относительную точности (погрешности) решений в обоих случаях.
3) Выведите решение (кажется, оба должны совпадать?) в текстовый файл в формате с плавающей запятой, каждый элемент решения в новой строке пронумерован, например,

   1-я координата 2.e-1
   2-я координата 3.14e0
   ….

Задание для конструирования пользователем системы, а также её решения с помощью факторизации матрицы и решателя linsolve.

4) Конструирование системы с симметричной матрицей.
   c) *выберите высокий порядок (n>1000) матрицы* `A` *и используйте для её построения функцию* `rand` *или* `randi`.
   d) *На основе вашей матрицы* A, *сконструируйте симметричную матрицу S докажите, что S является симметричной.*

Для матрицы S вычислите вектор-столбец `b` системы `Sx=b` (*столбец свободных членов вы уже умеете строить!*). Решите систему с опцией, предусматривающей симметрию матрицы системы

```
x=linsolve(P,b,opts)
```

Оцените время решения и сравните с встроенным в ML методом Гаусса, определите время алгоритма поиска решения, используйте функции в Приложении3.

Замечание 1. Пусть tG – время решения системы методом Гаусса, tSym – с QR факторизацией с помощью `linsolve`.

Время решений для сравнения – строка ML:

['tG=', num2str(tG), ' tSym=', num2str(tSym)]

В конце файла со всеми заданиями для пяти команд, вы обнаружите Приложения (целевой Help). Стоит в него заглянуть!

Команда 3.

1) Загрузите двоичный файл с данными V3.mat.
   a) Определите структуру матриц A и B визуально, их свойства (определитель, симметрию, предпосылки переходить к решению систем в классе разреженных матриц).
   b) Определите максимальный элемент в последнем блоке, блочно-диагональной матрицы. Найдите количество элементов полной матрицы, которые равны максимальному.
   c) Любым способом постройте вектор решения (одинаковый для обеих матриц A и B), вычислите произведение матриц на столбец решения, получите столбцы свободных членов a и b для обеих линейных систем.
2) Решите обе системы, выберите мотивировано, какую из них решать встроенным методом Гаусса, а какую на основе QR-факторизации. Определите абсолютную и относительную точности решений в обоих случаях.
3) Выведите решение (кажется, оба должны совпадать?) в текстовый файл в формате с плавающей запятой, каждый элемент решения в новой строке пронумерован, например,

   x1=   2.e-1
   x2=   3.14e0 и т.п.
   ….

Задание для конструирования пользователем системы, а также решения её с помощью факторизации матрицы и решателя linsolve.

(см. Приложения1 и 2)

1) Придумайте пример системы линейных алгебраических уравнений в матричной форме Ax=b (см. Приложение 1.)
   a) выберите высокий порядок (n>1000) матрицы `A` *и используйте для её построения функцию* `rand` *или* `randi`.
      - *реализуйте LU факторизацию, покажите (докажите) справедливость факторизации;*
      - *графически (визуально) представьте структуру полученных матричных множителей.*

   b) на основе вашей матрицы A, *сконструируйте симметричную матрицу* S. *Для неё*
      - *реализуйте QR факторизацию, покажите (докажите) справедливость факторизации;*
      - *графически (визуально) представьте структуру полученных матричных множителей.*

2) Используя положительно определенную матрицу `P` с положительными коэффициентами, симметричную и с диагональным преобладанием, постройте систему `Px=b`, высокого порядка (*столбец свободных членов вы уже умеете строить!*). Решите <u>эту систему</u> трижды с различными опциями, с помощью

```
linsolve(P,b,opts)
```

с опцией по умолчанию для матрицы системы P*, а также с опциями для симметричной и положительной определённой матрицы систем. Оцените время решения в каждом случае и сравните эффективность факторизации.* *<u>Заметим, что участвует эксперименте одна и та же положительно-определённая матрица, меняем лишь опции.</u>*

<u>Замечание 1</u>. Пусть tG – время решения системы методом Гаусса, tSym – время решения с QR факторизацией; tPosDef с факторизацией Холецкого. Время решений для сравнения – строка ML:

['tG=', num2str(tG), ' tSym=', num2str(tSym), ' tPosDef= ', num2str(tPosDef)]

<u>Замечание 2.</u> определите время алгоритма поиска решения, используйте функции, предлагаемые в `Приложении3`.

<u>Команда 4.</u>

В конце файла со всеми заданиями для пяти команд, вы обнаружите Приложения (целевой Help). Стоит в него заглянуть!

1) Загрузите двоичный файл с данными V4.mat.Определите элементы Workspace, их свойства (визуально – структуру матриц, определитель, симметрию, размах значений элементов матриц, найдите среднее значений размаха, процент ненулевых элементов матриц, объясните, зачем это может быть интересно).
   b) Выберите несимметричную матрицу. Проредите её бОльший блок в шахматном порядке, не используя циклы. Проверьте вырожденность.
   c) Задайте любым способом вектор решения (одинаковый для обеих матриц A и B), вычислите произведение матриц на вектор решения, получите столбцы свободных членов a и b для обеих линейных систем.
2) Решите систему, которая является симметричной, методом qr-факторизации, а несимметричную – методом Гаусса с использованием lu-факторизации, см. Приложение 1. Определите абсолютную и относительную точности решений в обоих случаях.
3) Выведите решение (кажется, оба должны совпадать?) в текстовый файл в формате с плавающей запятой, каждый элемент решения в новой строке пронумерован, например,

   1-й элемент   2.e-1
   2-й элемент   3.14e0
   ….

<u>Задание для конструирования пользователем системы, а также её решения с помощью факторизации матрицы и решателя linsolve.</u>

4) Конструирование системы с симметричной матрицей.
   e) *выберите высокий порядок (n>1000) матрицы* `A` *и используйте для её построения функцию* `randn`*, модифицируя матрицу так, чтобы она имела положительные элементы*.
   f) *На основе вашей матрицы* A*, сконструируйте симметричную матрицу* S*, докажите, что матрица S является симметричной.*

Для матрицы S вычислите вектор-столбец `b` системы `Sx=b` (*столбец свободных членов вы уже умеете строить!*). Решите систему с опцией, предусматривающей симметрию матрицы системы

```
x=linsolve(P,b,opts)
```

Оцените время решения и сравните с временем решения встроенного в ML метода Гаусса.

<u>Замечание 1</u>. Пусть tG – время решения системы методом Гаусса, tSym – с QR факторизацией с помощью `linsolve`.

Время решений для сравнения – строка ML:

['tG=', num2str(tG), ' tSym=', num2str(tSym)]

<u>*Замечание 2.*</u> *определите время алгоритма поиска решения, используя функции, предлагаемые в* Приложении3.

В конце файла со всеми заданиями для пяти команд, вы обнаружите Приложение Help (целевой). Стоит в него заглянуть!

Команда 5.

1) Загрузите двоичный файл с данными V5.mat.
   a) Определите структуру матриц A и B визуально, их свойства (определитель, симметрию, максимальные и минимальные значения матриц, количество ненулевых элементов).
   b) Последний блок полной несимметричной матрицы «передвиньте» в левый нижний угол.
   c) Любым способом постройте вектор решения симметричной матрицы, вычислите произведение матрицы на столбец решения, получите столбец свободных членов.
2) Решите полученную систему встроенным методом Гаусса, и на основе QR-факторизации. Определите абсолютную и относительную погрешности решений в обоих случаях.
3) Выведите решение в текстовый файл в машинном формате g, по три элемента в каждой строке

Задание для конструирования пользователем системы, а также решения её с помощью факторизации матрицы и решателя linsolve.

(см. Приложения1 и 2)

4) Придумайте пример системы линейных алгебраических уравнений в матричной форме Ax=b (см. Приложение 1.)
   a) выберите высокий порядок (n>1000) матрицы `A` *и используйте для её построения функцию* `rand` *или* `randi`.
      - *реализуйте LU факторизацию, покажите (докажите) справедливость факторизации;*
      - *графически (визуально) представьте структуру полученных матричных множителей.*
   b) на основе вашей матрицы A*, сконструируйте симметричную матрицу* S*. Для неё*
      - *реализуйте QR факторизацию, покажите (докажите) справедливость факторизации;*
      - *графически (визуально) представьте структуру полученных матричных множителей.*
3) Используя положительно определенную матрицу `P` с положительными коэффициентами, симметричную и с диагональным преобладанием, постройте систему `Px=b`, высокого порядка (*столбец свободных членов вы уже умеете строить!*). Решите эту систему трижды с различными опциями, с помощью

```
linsolve(P,b,opts)
```

с опцией по умолчанию для матрицы системы P*, а также с опциями для симметричной и положительной определённой матрицы системы. Оцените время решения в каждом случае и сравните эффективность факторизации.* *Заметим, что участвует эксперименте одна и та же положительно-определённая матрица, меняем лишь опции.*

Замечание 1. Пусть tG – время решения системы методом Гаусса, tSym – время решения с QR факторизацией; tPosDef с факторизацией Холецкого. Время решений для сравнения – строка ML:

['tG=', num2str(tG), ' tSym=', num2str(tSym), ' tPosDef= ', num2str(tPosDef)]

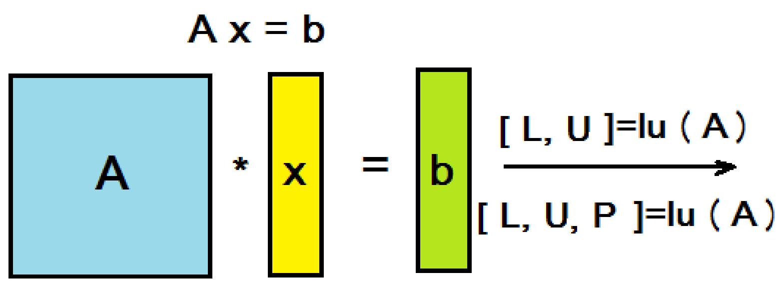
Замечание 2. определите время алгоритма поиска решения, используйте функции, предлагаемые в Приложении3.

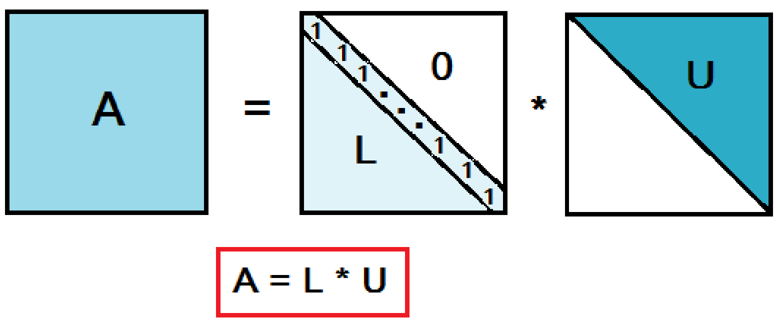
Приложение 1

Виды факторизации системы:

I) LU факторизация при решении СЛАУ

[L,U]=lu(A) (матрица A –n-порядка)





или для "психологически" треугольной

P * A = L * U, P - матрица перестановок

Решение методом Гаусса находится в MaLab:
x= A\b

MatLab реализация:

Ax=b (1)

Факторизация:

[L,U]=lu(A) % **A=LU**,

%% (1) преобразуется к виду:

LUx=b; % (2)

%% Обозначим **Ux≡y**, (3)

%%→ (2) преобразуем к с-ме

Ly=b

%% и найдем её решение $\mathbf{y}^*$:

$$\mathbf{y}^* = \mathbf{L}^{-1}\mathbf{b},$$

% решается прямой прогонкой:

$y_1{}^* = b_1, y_2{}^* = \%\ b_2 - l_{21} y_1{}^*$

$y_n{}^* = b_n - \sum_{i=1}^{n-1} l_{ni}\ \%\ y_i^*)$

%% Получим, учитывая(3):

$\mathbf{Ux = y^*}$, $\underline{\mathbf{x=}}\mathbf{U^{-1}\, y^* =}\ \underline{\mathbf{U^{-1}\, L^{-1} b}}$

%% это решение на основе

%% факторизации,

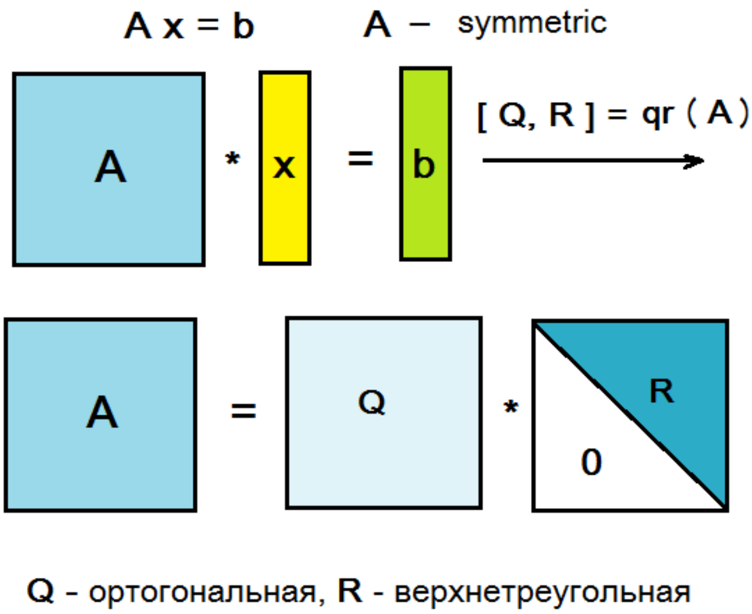
%% находится обратной

%% прогонкой:

$x_n = y_n^*/u_{n\,n},\ x_{n-1} = (y_{n-1}^* - u_{n-1\,n} x_n)/u_{n-1\,n-1}, \dots,$

$x_1 = (y_1^* - \sum_{i=2}^{n-1} u_{1i} x_1)/u_{1\,1}$

II) QR факторизация симметричной матрицы [Q,R]=qr(A)



$QQ^T = E \rightarrow Q^{-1} = Q^T$

MatLab реализация:

Ax=b

[Q,R]=qr(A)

% QR x=b, (4)

%% обозначим **R x ≡y**, (5)

%% (4) преобразуется к виду:

%% **Q y=b** (6)

%% Найдём её $\mathbf{y}^*$ решение:

$\mathbf{y^* = Q^T b}$ %т.к.$\mathbf{Q^{-1} = Q^T}$

%Согласно (5) $\mathbf{Rx = y^*} \rightarrow$

$\mathbf{x = R^{-1}\, y^*}$ %= $\mathbf{R^{-1}\, Q^T b}$

%% Решение на основе QR–факторизации:

$\underline{\mathbf{x = R^{-1}\, Q^T b}}$

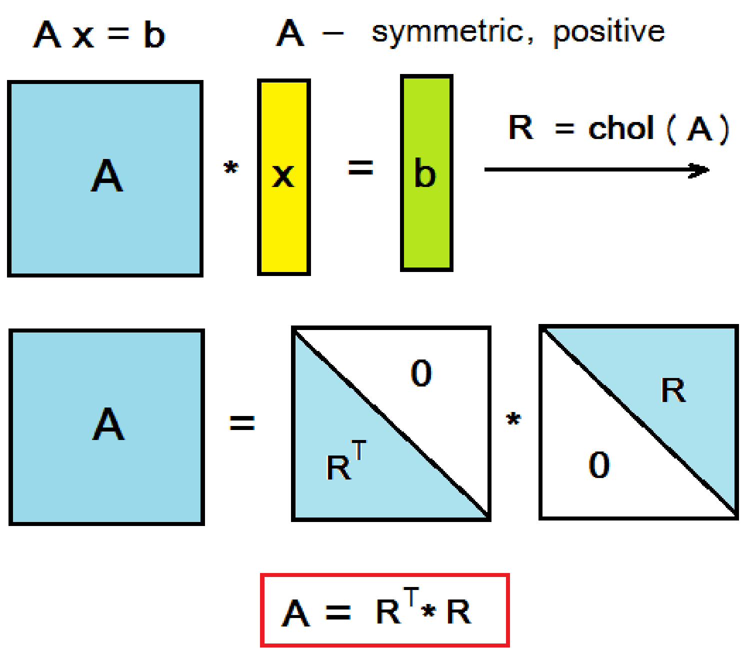
Как создать симметричную матрицу в ML?

Замечание. Используем один из критериев: Матрица **A** – положительно определенная (positive), если **A** *симметричная,* $a_{i,j}>0$*, и все собственные значения* $\lambda_i>0$.

В MatLab это легко проверяется: *all(eig(A)>0))→true*

III) Факторизация Холецкого — **MatLab реализация:**

R=chol(A)



%% С-ма Ax=b , A-симм. (7)
R=chol(A)
%A=R^TR, R верхнетреуголь-
%ная. Факторизованная
%% система (7): $\mathbf{R^TR\ x=b}$. **(8)**
%% Обозначив $\mathbf{Rx\equiv y}$, **(9)**
%%получим вместо (8) с-му
%% $\mathbf{R^Ty=b}$, и её решение
$\mathbf{y^*=(R^T)^{-1}\ b}$.
%%Согласно (9), $\mathbf{Rx=y^*}$ →
$\mathbf{x=R^{-1}\ y^*}$.
%%В результате решение (7)
%%на основе факторизации:
<u>$\mathbf{x=R^{-1}\ (R^T)^{-1}\ b}$</u>

<u>Замечание.</u> $\mathbf{R^T}\equiv R'$ *в MatLab*

Приложение 2

Решение линейных систем Ax=b на основе факторизации с помощью функции: x=linsolve(A,b,opts)

В поле зрения наших интересов факторизации LT, UT, SYM, POSDEF.

(см. расшифровку в таблице)

LT	UT	UHESS	SYM	POSDEF	RECT	TRANSA
true	false	false	false	false	true/false	true/false
false	true	false	false	false	true/false	true/false
false	false	true	false	false	false	true/false
false	false	false	true	true/false	false	true/false
false	false	false	false	false	true/false	true/false

Field Name	Matrix Property
LT	Lower triangular
UT	Upper triangular
UHESS	Upper Hessenberg
SYM	Real symmetric or complex Hermitian
POSDEF	Positive definite
RECT	General rectangular
TRANSA	Conjugate transpose — specifies whether the function solves A*X = B or A'*X = B

Важно! На основе факторизации описанной в Приложении пункта I): остаётся `opts` по умолчанию, метод Гаусса;
пункта II): задаем `opts.SYM=true`
пункта III): задаем две опции `opts.POSDEF=true`, `opts.SYM=true`

После задания опций решаем систему **Ax=b**, используя функцию **x=linsolve(A,b,opts)**

Приложение 3

Как работают временные скобки

A. tic toc

clear
tic
SYS=rand(10000,10001)/10000;
A=SYS(:,1:end-1);
x=SYS(:,end);
b=A*x; x=A\b
ouraccuracy=norm(b-x)
toc
% ouraccuracy = 0.0038
% Elapsed time is **8.612405** seconds
	clear
	tic
	SYS=rand(10000,10001)/10000;
	A=SYS(:,1:end-1);
	x=SYS(:,end);
	b=A*x; x=A\b;
	ouraccuracy=norm(b-x)
	toc
	% ouraccuracy = 0.0038
	% Elapsed time is **7.642928** seconds.

Назовите причину разницы во времени?

B. функция **cputime** может быть рассмотрена по желанию

```
tstart=cputime
SYS=rand(10000,10001)/10000;
   A=SYS(:,1:end-1);  x=SYS(:,end);
b=A*x;   x=A\b;
ouraccuracy=norm(b-x)
  TimeProcess= cputime-tstart
```