MATLAB Lecture2 -- Solving Linear Systems of Equations

线性方程组求解

Ref: MATLAB→Mathematics→Matrices and Linear Algebra →Solving Linear Systems of Equations

- **Vocabulary:**

coefficient matrix 系数矩阵	linear Systems of Equations 线性方程组
row elementary transpositions 行初等变换	basis 基
backslash 反斜线符号	least squares solution 最小二乘解
nonsingular matrix 非奇异阵，可逆矩阵	particular solution 特解
homogeneous system 导出组	solution space 解空间
linear homogeneous equation 线性齐次方程，齐次一次方程	
non-homogeneous system 非齐次系统（线性方程组）	
linearly independent 线性无关	pseudoinverse 广义逆
rational 有理数	component 元素
determinant 行列式	rank 秩
overdetermined system 超定系统（即方程组无精确解）	
underdetermined system 不定系统（即方程组有无穷多解）	
orthonormal basis 正交基	null 零空间，核空间

- **Some operations and functions**

  ' .' \ rank det inv rref null

- **Application on solving linear systems of equations** **Ax=b, A is an m×n matrix**

  ✧ Review:

  ✓ **b=0**

  Theory

  r(A) =n, there is only one exact solution zero.

  r(A) <n, there are infinite nonzero solutions.

  Computation

  r(A) =n, only zero is its exact solution

  r(A) <n, row elementary transpositions→basis for the null space of A

  ✓ **b≠0**

  Theory

  r(A)=r(A b)=n, there is only one exact solution

  r(A)=r(A b)<n, there are infinite solutions

  r(A)≠r(A b), there is no exact solution

  Computation

  m=n & r(A)=r(A b)=n, Cramer' rule; $A^{-1}b$; row elementary transposition

  m≠n & r(A) =n, row elementary transpositions

  m≠n & r(A)=r(A b)<n basis for the null space of A, particular solution

✧ MATLAB

MATLAB solves such linear equations without computing the inverse of the matrix.

The two division symbols, slash, /, and backslash, \, are used for the two situations where the unknown matrix appears on the left or right of the coefficient matrix.

X = A\B Denotes the solution to the matrix equation AX = B.
X = B/A Denotes the solution to the matrix equation XA = B.

The coefficient matrix A need not be square. If A is m-by-n, there are three cases.

m = n Square system. Seek an exact solution.
m > n Overdetermined system. Find a least squares solution.
m < n Underdetermined system. Find a basic solution with at most m nonzero components.

The backslash operator employs different algorithms to handle different kinds of coefficient matrices. The various cases, which are diagnosed automatically by examining the coefficient matrix, include:

Permutations of triangular matrices
Symmetric, positive definite matrices
Square, nonsingular matrices
Rectangular, overdetermined systems
Rectangular, underdetermined systems

General Solution

The general solution to a system of linear equations AX = b describes all possible solutions. You can find the general solution by:

Step 1. Solving the corresponding homogeneous system AX = 0. Do this using the null command, by typing null(A). This returns a basis for the solution space to AX = 0. Any solution is a linear combination of basis vectors.

Step 2. Finding a particular solution to the non-homogeneous system AX = b.

You can then write any solution to AX = b as the sum of the particular solution to AX = b, from step 2, plus a linear combination of the basis vectors from step 1.

Nonsingular Coefficient Matrix (can be verified by det or rank functions)

```
>> A = pascal(3);     % Obtain a Pascal matrix
>> u = [3; 1; 4];
>> x = A\u     % Try x=pinv(A)*u; x=inv(A'*A)*A'*u
x =
```

```
    10
   -12
     5
```

Singular Coefficient Matrix

If A is singular, the solution to AX = B either does not exist, or is not unique. The backslash operator, A\B, issues a warning if A is nearly singular and raises an error condition if it detects exact singularity.

If A is singular and AX = b has a solution, you can find a particular solution that is not unique, by typing

```
>> P = pinv(A)*b
```

pinv(A) is a Moore-Penrose pseudoinverse of A. The Moore-Penrose pseudoinverse is a matrix B of the same dimensions as A' satisfying four conditions:

A*B*A = A
B*A*B = B
A*B is Hermitian
B*A is Hermitian

The computation is based on svd(A).

If AX = b does not have an exact solution, pinv(A) returns a least-squares solution. $\text{pinv}(A)=(A'A)^{-1}A'$

For example,

```
>> A = [ 1     3     7
        -1     4     4
         1    10    18 ]
```

is singular, as you can verify by typing

```
>> det(A)
ans =
     0
```

Exact Solutions. For b =[5;2;12], the equation AX = b has an exact solution, given by

```
>> pinv(A)*b
ans =
    0.3850
   -0.1103
    0.7066
```

You can verify that pinv(A)*b is an exact solution by typing

```
>> A*pinv(A)*b
ans =
```

```
    5.0000
    2.0000
   12.0000
```

Least Squares Solutions. If b = [3;6;0], then AX = b does not have an exact solution. In this case, pinv(A)*b returns a least squares solution. If you type

```
>> A*pinv(A)*b
ans =
   -1.0000
    4.0000
    2.0000
```

you do not get back the original vector b.

You can determine whether AX = b has an exact solution by finding the row reduced echelon form of the augmented matrix [A b]. To do so for this example, enter

```
>> rref([A b])
ans =
    1.0000         0    2.2857         0
         0    1.0000    1.5714         0
         0         0         0    1.0000
```

Since the bottom row contains all zeros except for the last entry, the equation does not have a solution. In this case, pinv(A) returns a least-squares solution.

Overdetermined systems

Similar to **Least Squares Solutions**. Omitted.

Underdetermined Systems

```
>> R = fix(10*rand(2,4))     % Obtain an integer matrix by rand and fix function
R =
     6     8     7     3
     3     5     4     1
>> b = fix(10*rand(2,1))
b =
     1
     2
>> format rat     % display the solution in rational format
>> x0 = R\b       % finds a basic solution, which has at most m nonzero components
x0 =
       0
     5/7
       0
   -11/7
```

```
>> Z = null(R,'r')    % find an orthonormal basis (正交基) for the null space of R
Z =
      -1/2        -7/6
      -1/2         1/2
         1           0
         0           1
```

It can be confirmed that R*Z is zero and that any vector x where

```
>> syms k1 k2 k;    % Define three symbol variables
>> k=[k1; k2];
>> x = x0 + Z*k     %General solutions
```

Column Full Rank Systems

```
>> x = A\b
>> x = pinv(A)*b
>> x = inv(A'*A)*A'*b
```

theoretically computes the same least squares solution x, although the backslash operator does it faster.

A does not have full rank

```
x = A\b     %a basic solution; it has at most r nonzero components, where r is the rank of A.
x = pinv(A)*b     %the minimal norm solution because it minimizes norm(x).
x = inv(A'*A)*A'*b     %fails because A'*A is singular.
```

Example

```
>> A=fix(10*rand(3));
>> b=fix(10*rand(3,1));
>> ABar=[A b];
>> d=rref(ABar)     %The last column of d is the solution of system

d =

     1      0      0      -1/8
     0      1      0      73/48
     0      0      1     119/48
```