Computer-based Engineering Mathematics

Systems of linear equations in MATLAB - Introduction

Dr. Claudia Weis
Faculty of Engineering

IW
Wir machen Ingenieure

In this section: systems of linear equations

The simplest model in applied mathematics is a system of linear equations.

It is also by far the most important.

GILBERT STRANG*

*MathWorks Professor of Mathematics at the Massachusetts Institute of Technology (MIT).

📖 https://en.wikipedia.org/wiki/Gilbert_Strang

📖 http://www-math.mit.edu/~gs/

Refresher (1): System of linear equations and its matrix notation

- System of linear equations:

$$\begin{aligned} a_{11}x_1 &+ a_{12}x_2 &+ \ldots &+ a_{1n}x_n &= b_1 \\ a_{21}x_1 &+ a_{22}x_2 &+ \ldots &+ a_{2n}x_n &= b_2 \\ \vdots & \quad \vdots & & \quad \vdots & \quad \vdots \\ a_{m1}x_1 &+ a_{m2}x_2 &+ \ldots &+ a_{mn}x_n &= b_m. \end{aligned}$$

- Matrix notation:

(1.4.2) $\mathbf{Ax} = \mathbf{b}$,

where **A**, **x**, **b** are the following matrices.:

(1.4.3) $$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & \ldots & a_{2n} \\ . & . & & & . \\ . & . & & & . \\ a_{m1} & a_{m2} & \ldots & \ldots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_m \end{bmatrix}.$$

from: Prof. Gottschling, Maths 1 (ISE)

Refresher (2): important terms and facts

- matrix of coefficients $\boldsymbol{A}$
- homogenious system if $\boldsymbol{b} = \boldsymbol{0}$
- inhomogenious systems if $\boldsymbol{b} \neq \boldsymbol{0}$
- augmented matrix of the system

(1.3.5) $$\mathbf{A_b} = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array}\right]$$

is called the ***augmented matrix*** of the system (1.4.1).

- solution: set of real numbers $\{x_1, x_2, \dots, x_n\}$ that satisfies all the m equations
- general solution: a solution depending on parameters $\lambda_i \in \mathbb{R}$

- The rank of the augmented matrix and the rank of the matrix of coefficients determine whether the system of equations $\boldsymbol{Ax} = \boldsymbol{b}$ does or does not have a solution.

📖 Proposition 1.4.1 Maths 1 (ISE), Prof. Gottschling, UDE
📖 Lothar Papula: Mathematik für Ingenieure und Naturwissenschaftler Band 2, http://dx.doi.org/10.1007/978-3-658-07790-7 (*)
📖 Jörg Liesen, Linear Algebra, Springer, http://dx.doi.org/10.1007/978-3-319-24346-7 (*)
(*) Both books are available via UB from within the universtiy network (or VPN)

Refresher (3): more vocabulary

- consistence system: $\boldsymbol{Ax} = \boldsymbol{b}$ has a solution
- inconsistence system: $\boldsymbol{Ax} = \boldsymbol{b}$ does not have a solution
- over determined: $m > n$, i.e. it has more equations than unknowns
- determined: $m = n$, i.e the number of equations equals to the number of unknowns
- under determined: $m < n$, i.e. it has fewer equations than unknowns

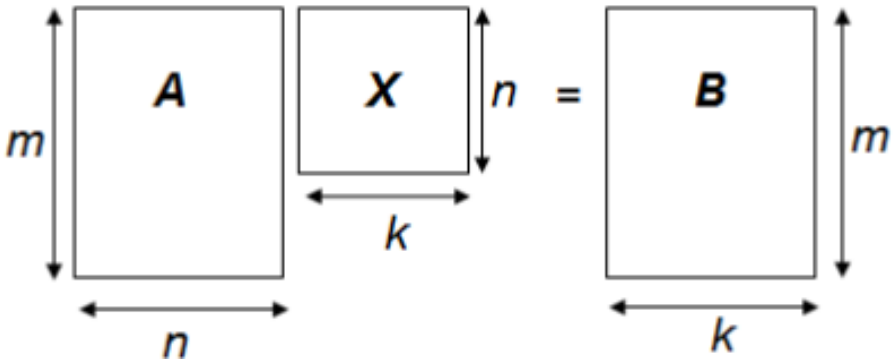
Popular solution strategies

- Gauss elimination
  - standard method of great importance for solving linear systems in practice
  - systematic elimination process
  - ➢ row operations on the augmented matrix reduce it to an echelon form
  - ➢ obtain the values of the unknowns by back substitution
- Solution with inverse matrix
  - direct calculation, if $\boldsymbol{A}$ is a regular matrix
  - ➢ multiply both sides of the linear equation $\boldsymbol{Ax} = \boldsymbol{b}$ with the inverse $\boldsymbol{A^{-1}}$
  - ➢ obtain solution of the linear system as $\boldsymbol{x} = \boldsymbol{A^{-1}}\,\boldsymbol{b}$
- Drawbacks of the two methods
  - explicit computation of the inverse of a regular matrix is time consuming and numerically not stable
  - Gauss elimination is numerically not stable
  - ➢ the two methods are not suitable to solve huge linear systems

Linear systems of equations in MATLAB

- Use the backslash operator for solving systems of linear equations.

- Solve k systems of equations to a given coefficient matrix $\boldsymbol{A}$ for the k column vectors of the right hand side matrix $\boldsymbol{B}$ with a single command.



Examples: $Ax_i = b_i$ vs. $AX = B$


```
>> A = round(10*rand(4))
A =
     2     5     1     8
     9    10    10     9
     2     1     0     1
     8     4     8     4

>> b1 = (1:4)'
b1 =
     1
     2
     3
     4

>> b2 = (5:8)'
b2 =
     5
     6
     7
     8
```

round(10*rand(4)) serves as a simple method to automatically generate an example matrix.

The rand command returns new numbers every time it is called.

possible solution 1

```
>> x1 = A\b1
x1 =
    1.6341
   -0.2927
   -1.0000
    0.0244

>> x2 = A\b2
x2 =
    3.5244
   -0.2805
   -2.5000
    0.2317
```

possible solution 2 (better using MATLAB's capabilities)

```
>> B = [b1, b2]
B =
     1     5
     2     6
     3     7
     4     8

>> X = A\B
X =
    1.6341    3.5244
   -0.2927   -0.2805
   -1.0000   -2.5000
    0.0244    0.2317
```

\-operator

- The \-operator chooses from different algorithms for the solution of linear systems

$$\boldsymbol{A}x = \boldsymbol{b}$$

- Which algorithm is selected depends upon the structure of the coefficiens matrix $\boldsymbol{A}$.

- MATLAB is able to recognize amongst others the following cases by analysis of $\boldsymbol{A}$:
  - Quadratic regular matrices
  - Over-determined systems
  - Under-determined systems
  - Triangular matrices

Goal for the next few lectures

- solve huge systems of linear equations
- consider specific properties in typical engineering problems
  - get an idea why we need to deal with huge systems
- learn about different numerical solution strategies for huge systems
  - sparse coefficient matrix
  - full coefficient matrix
  - memory and runtime issues

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

Questions?

- discussion forum @ Moodle
  (preferred, everyone can reply)
- e-mail to claudia.weis@uni-due.de
  (use your @uni-due.de email address!)