Linear systems
Part 2

Today
- Accuracy, sensitivity, condition number
- Norms
- Why \( A \backslash b \) more efficient than \( \text{inv}(A) \ast b \), i.e. Gaussian elimination more efficient than \( x = A^{-1} b \) ?

Accuracy
How accurate is the solution to \( Ax = b \)?
- Exact solution: \( x \) (not known)
- Computed solution: \( \hat{x} \)
- How big is the difference \( \| x - \hat{x} \| ? \)
- How can we estimate the difference without knowing \( x \) ?

Norms
First...
- We need a way to measure the size of vectors and matrices, corresponding to the absolute value for scalars
- We use norms, denoted \( \| \cdot \| \)
- Norms for both vectors and matrices, vector norm and matrix norm, respectively

Norms
- Vector norm
  Commonly used vector norms \( x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \)
  2-norm, euclidean norm:
  \[ \| x \|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2} \]
  1-norm, minimum norm:
  \[ \| x \|_1 = |x_1| + |x_2| + \cdots + |x_n| \]
  \( \infty \)-norm, maximum norm:
  \[ \| x \|_\infty = \max_i \{|x_1|, \ldots, |x_n|\} \]

Norms
- Different norms? Which is best?
- 1-norm:
- 2-norm:
**Informationsteknologi**

**Norm**
- From definition
  \[ \| A \| = \max_{i} \| A x_i \| \] , 1-norm, min. norm
  \[ \| A \|_\infty = \max_{j} \| A y_j \| \] , \( \infty \)-norm, max. norm
  \[ \| A \|_2 = \sqrt{\max \{ \lambda(A^T A) \}} \] , 2-norm, euclidian norm
  \[ \lambda(A^T A) \] is the eigenvalues of \( A^T A \)
- Usually 1- or \( \infty \)-norm as 2-norm is very expensive to compute

---

**Norms**
- Matrix norm
  Definition: \[ \| A \| = \max_{x \neq 0} \frac{\| A x \|}{\| x \|} \]
  Note, based on vector norms (\( A x \) and \( x \) are vectors)
- The definition not used for finding the norm of a specific matrix, but for deriving simpler formulas
- From definition you get formulas for matrix 1-norm, \( \infty \)-norm and 2-norm

---

**Norms**
- In Matlab
  - \( \text{norm}(x) \) 2-norm of vector \( x \)
  - \( \text{norm}(A) \) 2-norm of matrix \( A \)
  - \( \text{norm}(A,1) \) 1-norm
  - \( \text{norm}(A,\infty) \) \( \infty \)-norm

  ```matlab
  >> A =
   1 -2
   6  4
  >> norm(A,inf)
  ans =
   10
  >> norm(A,1)
  ans =
   7.2170
  ```

---

**Sensitivity, condition number**
**Ex 1)** Eq. system 1
\[
\begin{align*}
x_1 + x_2 &= 2 \\
x_1 - x_2 &= 0
\end{align*}
\]
Exact solution: \( x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \)

How differ:
\[
\begin{pmatrix}
x_1^* + x_2^* = 2 \\
\Delta x_1 - \Delta x_2 = \varepsilon
\end{pmatrix}
\]

\[
\begin{pmatrix}
\frac{1 + \varepsilon}{2} \\
\frac{1 - \varepsilon}{2}
\end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} \varepsilon \\ -\varepsilon \end{pmatrix} = x + \Delta x
\]

Perturbation out (in \( x \)) is of same order as perturbation in (in \( b \))

---

**Sensitivity, condition number**
**Ex 2)** Eq. system 2
\[
\begin{align*}
0.999x_1 - x_2 &= -0.001 \\
x_1 - x_2 &= 0
\end{align*}
\]
Exact: \( x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \)

How differ:
\[
\begin{pmatrix}
0.999\Delta x_1 + \Delta x_2 = -0.001 \\
\Delta x_1 - \Delta x_2 = \varepsilon
\end{pmatrix}
\]

\[
\begin{pmatrix}
\frac{1 - \varepsilon}{0.999} \\
1 - \frac{\varepsilon}{0.001}
\end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1000\varepsilon \\ 999\varepsilon \end{pmatrix} = x + \Delta x
\]

Perturbation out is magnified by factor 1000

---

**Sensitivity, condition number**
- Eq. system 2 is a sensitive, ill conditioned system/matrix
- Eq. system 1 is a well conditioned system – does not magnify errors in in data (the right-hand-side)
- Perturbation \( \varepsilon \) can be measuring error (accuracy in instruments) and roundoff errors
- Condition number does not depend on method used but on the nature of the underlying problem (e.i. the physical problem)
**Sensitivity, condition number**

- Ill condition systems have 'diffuse solution' – solution blurred by noise

**Condition number**

- Can derive following estimate of error in $x$

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \text{cond}(A) \frac{\|b - \hat{b}\|}{\|b\|}$$

$\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$ is the condition number of $A$

- Interpretation:
  - Rel. error in $x \leq$ cond. no.* Rel. error in in-data (right-hand-side) $b$
  - Noise in in-data, $b$ can be magnified with a factor $\text{cond}(A)$ in the comp. process

**Condition number**

- Condition number based on the different norms

- Ex 1

\[
A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{cond}_2(A) = 1 \\
\text{cond}_1(A) = 2 \\
\text{cond}_\infty(A) = 2
\]

- In Ex 1, noise in in-data does not magnify when solving the system (max a factor 2 in e.g. 1-norm), same order of magnitude

**Condition number**

- Ex 2

\[
A = \begin{pmatrix} 0.999 & -1 \\ 1 & -1 \end{pmatrix} \quad \text{cond}_2(A) = 3.9 \cdot 10^3 \\
\text{cond}_1(A) = 4 \cdot 10^3 \\
\text{cond}_\infty(A) = 4 \cdot 10^3
\]

- Noise in in-data, $t$ ex fel can be magnified with a factor $10^3$ when solving the system (agree with what we saw)

- But a cond. number $10^3$ is normally not a problem – must be related relative error in right-hand-side

**Condition number**

- Big condition number suggest that matrix close to singular

- In math a matrix is either singular non-singular - here a matrix can be almost singular

- Condition number
  - Depend on the nature of the underlying problem e.g. the physical reality might be sensitive to pertubations
  - Not affected by the algorithm/method that’s used
Condition number
- Condition number works as a warning – error might be big
- Best possible case: cond. Number = 1
  \[ \text{cond}(A) = \| A^{-1} \| \cdot \| A \| \geq \| A^{-1} A \| = 1 \]
  meaning no magnification at all.

The inverse
- Possible to solve \( Ax = b \) with \( x = A^{-1} b \)

```matlab
>> A = rand(10000,10000);
>> b = rand(10000,1);
>> tic; x = A\b; toc
Elapsed time is 85.4400 seconds.
>> tic; x = inv(A)*b; toc
Elapsed time is 251.89 seconds.
```
- Almost 3 times as long execution time

The inverse
- Usually avoid compute inverse as it is too expensive
- How's \( \text{cond}(A) = \| A \| \cdot \| A^{-1} \| \) calculated?

Usually estimates, for instance in Matlab's backslash

```matlab
>> x = A\b;
```
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate.

\[ \text{RCOND} = 8.113755e-020 \]

The inverse
- Inverse compute by solving \( AA^{-1} = I \),
equivalent with solving \( n \) right-hand-sides

\[
\begin{bmatrix}
1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 1
\end{bmatrix}
\]

\( n \) stycken kolonner
- LU factorize once, then \( n \) forward- and back substitution
- \( \frac{2}{3} n^3 \) forward- and back substitution
- For big \( n \)
- Almost 4 times more computations
- Take advantage of mainly zeros in right-hand-sides, so not so bad. But still about three times as long