How small is small?

- We have an intuitive notion of the size of a vector. Namely its **Euclidean length**

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

- The Euclidean length is also known as the **2-norm** (and hence the choice of notation above).
- Can we develop a similar concept for the size of a matrix?

Matrix Norm

- A matrix may be viewed as a *transformation* that maps a vector $\mathbf{x}$ onto a vector $A\mathbf{x}$.
- One possibility to measure the size of a matrix is to examine its effect on vectors; that is define

$$\|A\|_2 \stackrel{?}{=} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2}.$$

- The issue here is that the value of $\|A\|_2$ would depend on the vector $\mathbf{x}$.
- So we modify our attempted definition to give

$$\|A\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \left\{ \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \right\}$$

which will give us the inequality

$$\|A\mathbf{x}\|_2 \leqslant \|A\|_2 \, \|\mathbf{x}\|_2$$

for *all* $\mathbf{x}$.

Matrix Norm

- There is still a remaining issue, is this maximum *always finite*. If it is not then the inequality will not yield any useful information!
- Note that, for any non-zero scalar α,

$$\|\alpha\mathbf{x}\|_2 = |\alpha|\,\|\mathbf{x}\|_2 \qquad \text{and} \qquad A(\alpha\mathbf{x}) = \alpha A\mathbf{x}.$$

- Therefore, for any non-zero vector $\mathbf{x}$,

$$\frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \|A\hat{\mathbf{x}}\|_2 \qquad \text{where} \qquad \hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$$

  is a *unit* vector.
- Consequently we can now write

$$\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2.$$

- This now guarantees that $\|A\|_2$ will **always be finite** (for those who are mathematically inclined, since $\|\mathbf{x}\|_2 = 1$ is a *compact* set).

Errors & Residuals

- We want to estimate the (relative) error in the computed solution, $\tilde{\mathbf{x}}$ to the system $A\mathbf{x} = \mathbf{b}$.
- The error (using the Euclidean distance) is

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2$$

- However, has before, we cannot compute this since we do not know what the exact solution, $\mathbf{x}$.
- We can (easily!) compute

$$\tilde{\mathbf{b}} = A\tilde{\mathbf{x}}$$

  and then compute

$$\|\tilde{\mathbf{b}} - \mathbf{b}\|_2.$$

  This quantity is called the **residual**.
- Is there a relationship between the error (which is the quantity we want but cannot compute) and the residual (which we can easily compute)?

Errors & Residuals

- Now $\mathbf{x} = A^{-1}\mathbf{b}$ and so

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 = \|A^{-1}\tilde{\mathbf{b}} - A^{-1}\mathbf{b}\|_2 \leqslant \|A^{-1}\|_2 \, \|\tilde{\mathbf{b}} - \mathbf{b}\|_2.$$

- Furthermore

$$\|\mathbf{b}\|_2 = \|A\mathbf{x}\|_2 \leqslant \|A\|_2 \, \|\mathbf{x}\|_2$$

  and so

$$\frac{1}{\|\mathbf{x}\|_2} \leqslant \frac{\|A\|_2}{\|\mathbf{b}\|_2}.$$

- Combining these two inequalities, we have

$$\frac{\|\tilde{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leqslant \|A\|_2 \frac{\|\tilde{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{b}\|_2} \leqslant \|A\|_2 \, \|A^{-1}\|_2 \frac{\|\tilde{\mathbf{b}} - \mathbf{b}\|_2}{\|\mathbf{b}\|_2}.$$

Condition Number

- In other words

$$\text{relative error} \leqslant K(A) \times \text{relative residual}$$

  where

$$K(A) = \|A\|_2 \, \|A^{-1}\|_2$$

- $K(A)$ is called the **condition number** of A.
- This formula gives us an *upper bound* on the relative error in terms of the relative residual (a quantity that we can compute). One problem remains ...
- Can we compute the condition number? We certainly do not want to compute A^{-1} since this would involve another Gauss(-Jordan!) row reduction.

Matrix Norm – Computation

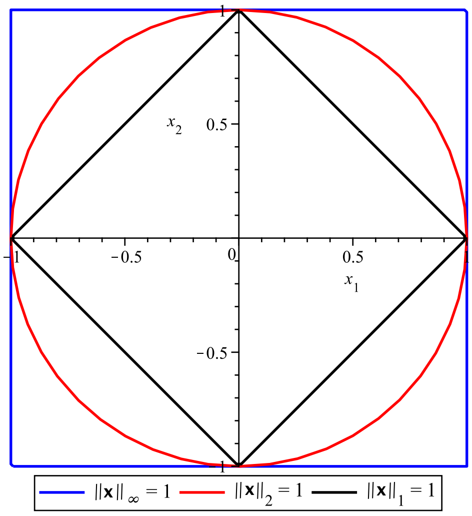
- Unfortunately $\|A\|_2$ is also **not** straightforward to compute (even for 2×2 matrices).
- Note
$$\|\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{x}.$$
- Consequently
$$\|A\mathbf{x}\|_2^2 = \mathbf{x}^T A^T A\mathbf{x}.$$
- This is a *quadratic form* and so it maximum value on the unit circle, $\|\mathbf{x}\|_2 = 1$, will be the largest eigenvalue of $A^T A$.
- In other words
$$\|A\|_2 = (\text{largest eigenvalue of } A^T A)^{1/2}.$$
- We certainly do *not* want to solve an eigenvalue problem (in fact, potentially two eigenvalue problems) to compute the condition number!

Vector Norms

- The above analysis is not specific to the Euclidean distance. If we can find other measures of distances then we might have an easier computational problem.
- One alternative to the Euclidean norm for vectors is the **1-norm**
$$\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$$
- In fact we can generalize this to the **p-norm**
$$\|\mathbf{x}\|_p = \left(|x_1|^p + |x_2|^p + \cdots + |x_n|^p\right)^{1/p}$$
(hence the alternative name for the Euclidean distance, the 2-norm).
- We can also take the “limit” as $p \to \infty$ to obtain
$$\|\mathbf{x}\|_\infty = \max\left\{|x_1|,\ |x_2|,\ \ldots,\ |x_n|\right\}$$
called, not surprisingly, the **infinity-norm**.

Vector Norms

- Of these norms, three are important in computational mathematics:
  - 1-norm
  - 2-norm
  - ∞-norm
- The *unit balls*, $\|\mathbf{x}\| = 1$, in these norms differ. In **R**2 we have



Matrix Norms

- *Any* vector norm $\|\cdot\|$ induces a norm of matrices via the construction above

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|.$$

  This norm is called the **operator norm** induced by $\|\cdot\|$
- Since the unit balls differ for different norms, it may be that the operator norm is easier to compute for some vector norms.

1-NORM

- Let the columns of A be denoted by the vectors $\mathbf{a}_i$; that is
$$A = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}.$$
- Let
$$M = \max_i \{\|\mathbf{a}_i\|_1\}$$
the largest absolute *column* sum.
- For $\|\mathbf{x}\|_1 = 1$ we have $|x_1| + |x_2| + \cdots + |x_n| = 1$ and so
$$\begin{aligned} \|A\mathbf{x}\|_1 &= \|x_1\,\mathbf{a}_1 + x_2\,\mathbf{a}_2 + \cdots + x_n\,\mathbf{a}_n\|_1 \\ &\leqslant |x_1|\|\mathbf{a}_1\|_1 + |x_2|\|\mathbf{a}_2\|_1 + \cdots + |x_n|[\|]\|1]\mathbf{a}_n \\ &\leqslant (|x_1| + |x_2| + \cdots + |x_n|)\,M = M \end{aligned}$$

- If the largest absolute column sum occurs in column k then, with $\mathbf{x} = \mathbf{e}_k$, the k^{th} unit vector,
$$\|A\mathbf{e}_k\|_1 = \|\mathbf{a}_k\|_1 = M.$$
- Therefore
$$\|A\|_1 = M = \max_j \sum_i |a_{ij}| = \text{largest absolute column sum.}$$

MATRIX NORMS

- By a similiar argument
$$\|A\|_\infty = \max_i \sum_j |a_{ij}| = \text{largest absolute row sum.}$$
- Therefore both the 1-norm and ∞-norm are simple to compute (and this is why they are important in computational mathematics).
- The other p-norms are even harder to compute than the 2-norm.
- The MATLAB command `norm` will only compute the 1-, 2- or ∞-norm of a matrix. It defaults to the 2-norm.

Example Revisited

- In our earlier example

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4.01 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix}$$

and the computed solution was

$$\tilde{\mathbf{x}} \approx \begin{bmatrix} 0.133 \\ 0.1 \end{bmatrix}.$$

- In this case we can compute

$$A^{-1} = \begin{bmatrix} 401 & -200 \\ -200 & 100 \end{bmatrix}$$

and so the condition number (using the 1-norm) is

$$K(A) = \|A\|_1 \, \|A^{-1}\|_1 = 6.01 \times 601 = 3612.$$

Thus the relative error could be as much as 3612 times the relative residual.

Example Revisited

- Now

$$\tilde{\mathbf{b}} = A\tilde{\mathbf{x}} = \begin{bmatrix} 0.333 \\ 0.667 \end{bmatrix} \qquad \text{and} \qquad \tilde{\mathbf{b}} - \mathbf{b} = \begin{bmatrix} 0.000333 \\ -0.000333 \end{bmatrix}.$$

- So

$$\text{relative residual} = \frac{\|\tilde{\mathbf{b}} - \mathbf{b}\|_1}{\|\mathbf{b}\|_1} = \frac{0.000667}{1} = 0.000667.$$

- Even though the relative residual is small, we have

$$\text{relative error} \leqslant 3612 \times 0.000667 = 2.41!$$

- The actual relative error is

$$\text{relative error} = \frac{\|\tilde{\mathbf{x}} - \mathbf{x}\|_1}{\|\mathbf{x}\|_1} = \frac{0.3}{\frac{1}{3}} = 0.9.$$

Condition Number

- The condition number is giving us a measure of the *degree of precision* needed for the computation.
- If we used k-digit arithmetic in the above example, the relative residual would be

$$\text{relative residual} = 0.6666...7 \times 10^{-k}.$$

- Therefore

$$\text{relative error} \leqslant 3612 \times 0.6666...7 \times 10^{-k} \leqslant 10^{-k+4}.$$

- That is *only the first $k - 4$ digits would be reliable*. In other words, to compute the solution with an accuracy of 1%, we would need to use at least 6-digit arithmetic.

Matlab and Condition Number

- The MATLAB command `cond` computes the condition number (it defaults to the 2-norm).
- MATLAB also has a command `rcond`. This command *estimates* the reciprocal condition number (using the 1-norm); that is

$$\texttt{rcond}(\texttt{A}) \approx \frac{1}{\|A\|_1 \, \|A^{-1}\|_1}.$$

- The rationale for giving the reciprocal is that if

$$\frac{1}{K(A)} \approx 10^{-q}$$

then the *last q digits are unreliable*. Since MATLAB, by default uses 16-digits, this means that the first $16 - q$ digits are reliable.
- When using "matrix division" (that is, the \ operator), MATLAB will use `rcond` to check for (and warn about) ill-conditioned systems.

Scaling Revisited

- In the scaling example, the badly scaled version of the equations had a coefficient matrix
$$A = \begin{bmatrix} 1 & -0.0001 \\ 100 & 1 \end{bmatrix}$$
whereas the scaled version was
$$A_{\text{scaled}} = \begin{bmatrix} 1 & -0.01 \\ 1 & 1 \end{bmatrix}.$$
- Using `rcond` we have

```
>> A=[1 -0.0001;100 1];
>> rcond(A)
ans =
  9.9010e-005

>> Ascaled=[1 -0.01;1 1];
>> rcond(Ascaled)
ans =
    0.3807
```

- Thus, in the badly scaled case, 4 digits are lost whereas only 1 digit is lost in the scaled case.