Useful properties of matrix & vector norms

3/24/11 L

We previously saw that

$$\|Ax\| \leq \|A\| \cdot \|x\| \qquad (1)$$

for any matrix A, and any vector x (of dimensions $m \times m$ and $m \times 1$, respectively).

Note that, when writing an expression such as (1), the matrix norm $\|A\|$ is understood to be the inferred norm from the vector norm used in $\|Ax\|$ and $\|x\|$. Thus

$$\|Ax\|_1 \leq \|A\|_1 \cdot \|x\|_1$$

and

$$\|Ax\|_\infty \leq \|A\|_\infty \cdot \|x\|_\infty$$

are both valid, but we cannot mix and match, e.g.:

$$\|Ax\|_\infty \leq \|A\|_2 \cdot \|x\|_1$$ → NOT CORRECT

When solving a linear system $A\underline{x} = \underline{b}$, computer algorithms are only providing an approximation ($\underline{x}_{app}$) to the exact solution ($\underline{x}_{ex}$). This is due to factors such as finite precision, roundoff errors or even imperfect solution algorithms. In either case, we have an <u>error</u> (error vector, in fact) defined as

$$\underline{e} = \underline{x}_{ap} - \underline{x}_{ex}$$

Naturally, we would like to have an understanding of the <u>magnitude</u> of this error (e.g. some appropriate norm $\|\underline{e}\|$). The problem is that we do not know the exact, pristine solution $\underline{x}_{ex}$!

One remedy is offered via the <u>residual vector</u> defined as:

$$\underline{r} = \underline{b} - A\underline{x}_{app}$$

The vector $\underline{r}$ is something we can compute practically since it involves only known quantities ($\underline{b}, A, \underline{x}_{app}$).

Furthermore, we have:

$$\underline{r} = \underline{b} - A\,\underline{x}_{app}$$

$$= A\,\underline{x}_{ex} - A\,\underline{x}_{app}$$

$$= -A\left(\underline{x}_{app} - \underline{x}_{ex}\right)$$

$$= -A\,\underline{e} \quad \Longrightarrow \quad \underline{r} = -A\,\underline{e}$$

$$\underline{e} = -A^{-1}\,\underline{r}$$

The last equation links the error with the residual.

Furthermore, we can write

$$\|\underline{e}\| = \|A^{-1} r\| \leq \|A^{-1}\|\,\|r\|$$

This equation provides a <u>bound</u> for the error, as a function of $\|A^{-1}\|$ and the norm of the computable vector $\underline{r}$! Note that:

→ We can obtain this estimate <u>without</u> knowing the exact solution, <u>but</u>

→ We need $\|A^{-1}\|$ and generally, computing A^{-1} is just as difficult (if not more) than finding $\underline{x}_{ex}$. <u>However</u> there are special cases where an estimate of $\|A^{-1}\|$ can be obtained.

A different source of error :

Sometimes, the right-hand-side (b) of $Ax = b$ has errors that make it deviate from its intended value. For example in the Vandermonde matrix method for polynomial interpolation, $\underline{b}$ contains the samples $(y_1 = f(x_1), y_2, \ldots, y_n)$ where $y_i = f(x_i)$. An error in a measuring device supposed to sample $f(x)$ could lead to erroneous readings y_i^* instead of y_i. In general, measuring inaccuracies can lead to the right-hand-side vector $\underline{b}$ being mis-represented as $\underline{b}^*$ $(\neq b)$.

In this case, instead of the intended solution $x = A^{-1}b$ we in fact compute $x^* = A^{-1}b^*$.

How important is the error $e = x^* - x$ that is caused by this misrepresentation of b?

Let us introduce some notation:

Let $\delta \underline{b} := \underline{b}^* - \underline{b}$

$\delta \underline{x} := \underline{x}^* - \underline{x}$

$$A\underline{x} = \underline{b}$$
$$A\underline{x}^* = \underline{b}^*$$

$$A(\underline{x}^* - \underline{x}) = \underline{b}^* - \underline{b}$$

$$A\,\delta\underline{x} = \delta\underline{b}$$

$$\delta x = A^{-1}\,\delta b$$

Taking norms:

$$\|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\|\,\|\delta b\| \qquad (1)$$

Thus the error in the computed solution δx is proportional to the error in b.

An even more relevant question is: How does the relative error $\frac{\|\delta x\|}{\|x\|} = \frac{\|x^* - x\|}{\|x\|}$ compare to the relative error in $\underline{b}$ $\frac{\|\delta b\|}{\|b\|}$? This may be more useful to know, since $\|\delta b\|$ may be impossible to compute (if we don't know the real $\underline{b}$!).

For this, we write

$$Ax = b \Rightarrow \|b\| = \|Ax\| \leq \|A\| \cdot \|x\|$$

$$\Rightarrow \frac{1}{\|x\|} \leq \|A\| \cdot \frac{1}{\|b\|} \quad (2)$$

Multiplying (1) & (2) we get:

$$\frac{\|\delta x\|}{\|x\|} \leq \underbrace{\|A\| \cdot \|A^{-1}\|}_{\kappa(A)} \cdot \frac{\|\delta\|}{\|b\|}$$

Thus the relative error in $\underline{x}$ is bounded by a multiple of the relative error in $\underline{b}$! The multiplicative constant $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ is called the condition number of A, and is an important measure of the sensitivity of a linear system $Ax=b$ to being solved on a computer, in the presence of inaccurate values.

e.g. If the relative error $\frac{\|\delta b\|}{\|b\|}$ is 0.0001%, but $\kappa(A) = 100.000$ (could happen!), Then we could have up to a 10% error in the computed $\underline{x}$!

Simply, almost <u>any</u> b will have <u>some</u>, small relative error, due to the fact it is represented on a computer up to machine precision! The relative error will be at least as much as the machine epsilon due to roundoff!

$$\frac{\|\delta b\|_\infty}{\|b\|} \geq \varepsilon \cong 10^{-7} \quad \text{(for } \underline{\text{floats}}\text{)}.$$

But, how bad can the condition number get?

<u>Very</u> bad, at times. For example:

Hilbert matrices $H_n \in \mathbb{R}^{n\times n}$ $(H_n)_{ij} = \dfrac{1}{i+j-1}$

$$H_5 = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{bmatrix}$$

$$\kappa_\infty(H_5) = \|H_5\|_\infty \cdot \|H_5^{-1}\|_\infty \cong 10^6 !$$

Thus, any attempt at solving $H_5 x = b$ would be subject to a relative error up to 10% <u>just due</u> to roundoff errors in <u>b</u>!

Another case : near-singular matrices

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 6+\varepsilon \end{bmatrix}$$

As $\varepsilon \to 0$ A becomes singular (non-invertible)

In this case, $\kappa(A) \to \infty$.

What is the best case for $\kappa(A)$?

Lemma For any vector-induced matrix norm, we have $\|I\| = 1$.

Proof From definition:

$$\|I\| = \max_{x \neq 0} \frac{\|Ix\|}{\|x\|} = \max_{x \neq 0} \frac{\|x\|}{\|x\|}$$

Using property (iv) of matrix norms, we get:

$$I = A \cdot A^{-1} \Rightarrow 1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\|$$

Thus $\boxed{\kappa(A) \geq 1}$

The "best" conditioned matrices are of the form $A = c \cdot I$!

And have $\kappa(A) = 1$.

Solving linear systems of equations in MATLAB

General case : Backslash operator \

If $x, y \in \mathbb{R}^n$ $A \in \mathbb{R}^{n \times n}$

and $Ax = y$

then $x = A^{-1}y$ is implemented in MATLAB

as $\boxed{x = A \backslash y}$

The actual algorithm used is selected "automatically" given the nature of A. Worst case, though, will require a run-time proportional to n^3! (We write: Cost=$O(n^3)$)

Alternative LU factorization.

Any non-singular matrix can be written as

$A = L \cdot U$ where $L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ l_{n1} & \cdots & l_{n,n-1} & l_{nn} \end{bmatrix}$ lower triangular

$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & \vdots \\ \vdots & & \ddots & u_{n-1,n} \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix}$ upper triangular.

In MATLAB :

$$[L,U] = lu(A); \quad \longrightarrow \text{Cost } O(n^3)$$

We then reduce $Ax=b$ to 2 triangular systems

$$Ax = b$$

$$LU\underbrace{x}_{z} = b$$

$$Lz = b \quad (1)$$

followed by $Ux = z$ (2)

In matlab :

$z = L \backslash b$ Cost $O(n^2)$!

$x = U \backslash z$ Cost $O(n^2)$!

Beneficial, if we need to solve several systems

$$Ax_1 = b_1$$
$$Ax_2 = b_2$$
$$\vdots$$
$$Ax_k = b_k$$

In which case the cost of factorization is paid only once!