

Notes for Math 221, Lecture 3, Jan 27, 2022

To summarize our approach to understanding accuracy in the face of roundoff (or other) error, our goal will be to prove that algorithms are "backward stable". For example if our algorithm for computing the scalar function $f(x)$ computes

$$\text{alg}(x) = f(x + \delta) \sim f(x) + f'(x) * \delta$$

where δ is "small", then relative error can be approximated by

$$\begin{aligned} |(\text{alg}(x) - f(x)) / f(x)| &\leq |f'(x) * \delta / f(x)| \\ &= |f'(x) * x / f(x)| * |\delta / x| \end{aligned}$$

Def: The factor $|f'(x) * x / f(x)|$ is called the `condition_number` of the function f evaluated at x .

We want to use the same approach for the problem of solving $A * x = b$, but where we get $(A + \Delta) * \hat{x} = b$ instead, where Δ is "small" compared to A , or the eigenproblem $A * x = \lambda * x$, but get

$$(A + \Delta) * \hat{x} = \lambda_{\text{hat}} * \hat{x}$$

where again Δ is "small" compared to A .

To formalize the notion of "small", we need to understand vector and matrix norms. In both cases, we may say

we want to compute $x = f(A)$, but get $\hat{x} = \text{alg}(A) = f(A + \Delta)$

So to bound the error, we write

$$\text{error} = \hat{x} - x = \text{alg}(A) - f(A) = f(A + \Delta) - f(A)$$

Assuming Δ is "small", and taking the first term of the Taylor expansion, we get

$$\text{error} \sim J_f(A) * \Delta \quad \text{where } J_f(A) \text{ is the Jacobian of } f$$

If A and x were scalars, we could take absolute values and get an absolute error bound:

$$|\text{error}| \lesssim |J_f(A)| * |\Delta|$$

and proceed as above.

In the cases most relevant to linear algebra, A and x are not scalars but matrices and vectors (with obvious generalizations, depending on the problem). To generalize this error bound, we need to generalize absolute values, which leads us to norms.

Goals: Matrix and vector norms

Singular Value Decomposition
Condition numbers for $Ax=b$

Matrix and vector norms

Def norm: Let B be linear space \mathbb{R}^n (or \mathbb{C}^n).

It is normed if there is a function $\|\cdot\|: B \rightarrow \mathbb{R}$ s.t.

(1) $\|x\| \geq 0$, and $\|x\| = 0$ iff $x=0$
(positive definiteness)

(2) $\|c * x\| = |c| * \|x\|$ (homogeneity)

(3) $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)

Examples: p -norm = $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$, for $p \geq 1$

Euclidean norm = 2-norm = $\|x\|_2$

Note: x real $\Rightarrow \|x\|_2^2 = \sum_i x_i^2 = x^T * x$
infinity-norm = $\|x\|_\infty = \max_i |x_i|$
C-norm = $\|C*x\|$ where C has full column rank
(Question 1.5)

Lemma (1.4): all norms are equivalent
i.e. given any $\text{norm}_a(x)$ and $\text{norm}_b(x)$, there are positive constants α and β such that
 $\alpha * \text{norm}_a(x) \leq \text{norm}_b(x) \leq \beta * \text{norm}_a(x)$
(proof: compactness)

This lemma is an excuse to use the easiest norm to prove later results.

Def: matrix norm (vector norm on $m \times n$ vectors)
(1) $\|A\| \geq 0$, and $\|A\| = 0$ iff $A=0$
(positive definiteness)
(2) $\|c*A\| = |c| * \|A\|$ (homogeneity)
(3) $\|A+B\| \leq \|A\| + \|B\|$ (triangle inequality)

Ex: max norm = $\max_{ij} |A_{ij}|$
Frobenius norm = $(\sum_{ij} |A_{ij}|^2)^{1/2}$

Def: operator norm
 $\text{norm}(A) = \max_{\{ \text{nonzero } x \}} \text{norm}(A*x) / \text{norm}(x)$

Lemma (1.6): The operator norm is a matrix norm.
proof: Question 1.15

Lemma (1.7): if $\text{norm}(A)$ is an operator norm, there exists x such that
 $\text{norm}(x)=1$ and $\text{norm}(A*x)=\text{norm}(A)$.
proof: $\text{norm}(A) = \max_{\{ \text{nonzero } x \}} \text{norm}(A*x) / \text{norm}(x)$
 $= \max_{\{ \text{nonzero } x \}} \text{norm}(A * x / \text{norm}(x))$
 $= \max_{\{ \text{unit vector } y \}} \text{norm}(A * y)$
 y attaining this maximum exists since $\text{norm}(A*y)$ is a continuous function of y on compact (closed and bounded) set = unit ball

Now we turn to orthogonal and unitary matrices, which we need for the SVD.

Notation: $Q^* = \text{conj}(Q^T)$,
sometimes we write Q^H (H stands for "Hermitian",
since a matrix satisfying $A=A^H$ is called Hermitian)

Def: orthogonal matrix: Q square, real and $\text{inv}(Q) = Q^T$
unitary matrices: Q square, complex, and $\text{inv}(Q) = Q^*$
(For simplicity, we state results for real case, but all extend to complex.)

Fact: Q orthogonal $\Leftrightarrow Q^T * Q = I \Leftrightarrow$ all columns mutually orthogonal

and are unit vectors, $Q*Q^T = I$ implies same about rows

Fact: $\text{norm}(Q*x,2) = \text{norm}(x,2)$ - aka Pythagorean theorem

Proof: $\text{norm}(Q*x,2)^2 = (Q*x)^T*(Q*x) = x^T*Q^T*Q*x = x^T*x = \text{norm}(x,2)^2$

Fact: Q and Z orthogonal $\Rightarrow Q*Z$ orthogonal

Proof: $(Q*Z)^T * (Q*Z) = Z^T*Q^T * Q*Z = Z^T * I * Z = I$

Fact: If Q is $m \times n$, with $n < m$, and $Q^T * Q = I_n$, then you can add $m-n$ columns to Q to make it $m \times m$ and orthogonal, so $Q^T*Q=I_m$ (there may be infinitely many ways to this, we can give a proof later, but this is a useful fact in some proofs)

Lemma (most proofs in homework, Q1.16)

(1) $\text{norm}(A*x) \leq \text{norm}(A)*\text{norm}(x)$ for vector and its operator norm

(2) $\text{norm}(A*B) \leq \text{norm}(A)*\text{norm}(B)$ for operator norm

(3) $\text{norm}(Q*A*Z,2) = \text{norm}(A,2)$ if Q, Z orthogonal

(4) $\text{norm}(Q,2)=1$

(5) $\text{norm}(A,2) = \text{sqrt}(\text{lambda_max}(A^T*A))$

(6) $\text{norm}(A^T,2) = \text{norm}(A,2)$

proof of (5) only:

$$\begin{aligned} \text{norm}(A,2) &= \max_{\text{nonzero } x} \text{norm}(A*x)/\text{norm}(x) \\ &= \max_{\text{nonzero } x} \text{sqrt}((A*x)^T*(A*x) / \text{sqrt}(x^T*x)) \\ &= \text{sqrt}(\max_{\text{nonzero } x} (A*x)^T*(A*x) / (x^T*x)) \\ &= \text{sqrt}(\max_{\text{nonzero } x} x^T*A^T*A*x / x^T*x) \end{aligned}$$

Use the fact that A^T*A is symmetric and so has eigenvalue decomposition:

$$A^T*A*q_i = \lambda_i * q_i$$

where λ_i real, q_i real, unit, orthogonal.

Let $Q = [q_1, \dots, q_n]$ and $\text{Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$, so that $A^T*A*Q = Q*\text{Lambda}$, and $A^T*A = Q*\text{Lambda}*inv(Q) = Q*\text{Lambda}*Q^T$
Continuing from above:

$$\begin{aligned} \text{norm}(A,2) &= \text{sqrt}(\max_{\text{nonzero } x} x^T*Q*\text{Lambda}*Q^T*x / x^T*x) \\ &= \text{sqrt}(\max_{\text{nonzero } x} x^T*Q*\text{Lambda}*Q^T*x / x^T*Q*Q^T*x) \\ &= \text{sqrt}(\max_{\text{nonzero } y} y^T*\text{Lambda}*y / y^T*y) \\ &\quad \text{where } y = Q^T*x \\ &= \text{sqrt}(\max_{\text{nonzero } y} \sum_i y_i^2 \lambda_i / \sum_i y_i^2) \\ &\leq \text{sqrt}(\max_{\text{nonzero } y} \lambda_{\text{max}} * \sum_i y_i^2 / \sum_i y_i^2) \\ &= \text{sqrt}(\lambda_{\text{max}}), \text{ attained by choosing } y_{\text{max}} = 1, \text{ rest } 0 \end{aligned}$$

SVD = Singular Value Decomposition

The SVD is a Swiss Army Knife of numerical linear algebra:

Given the SVD of A, one can easily

solve $Ax=b$ (when A is square),

solve an overdetermined or underdetermined least squares problem with rectangular A (whether A is full rank or not), compute the eigenvalues and eigenvectors of $A \cdot A^T$ and $A^T \cdot A$, or compute eigenvalues and eigenvectors of A if A is symmetric. Furthermore, one can use the SVD to write down error bounds for all these problems. It is more expensive to compute than other algorithms specialized for these problems, so it may not be the algorithm of first resort.

History: The first complete statement goes back to Eckart & Young in 1936. The first reliable algorithm was by Golub & Kahan in 1965, with faster ones since then, to be discussed in Chapter 5. Perhaps the fastest algorithm appears in the prize-winning 2010 PhD thesis by Paul Willems, which remains to be incorporated into LAPACK. (We have found some numerical examples on which the algorithm fails to be accurate enough, and have not managed to fix it yet, despite conversations with the author. So learning about and testing this algorithm is a possible (and challenging) class project.)

Thm. Suppose $A = m \times n$, then there is an orthogonal matrix $U = [u(1), \dots, u(m)]$
 diagonal matrix $\Sigma = \text{diag}(\sigma(1), \dots, \sigma(m))$
 with $\sigma(1) \geq \sigma(2) \geq \dots \geq 0$
 orthogonal matrix $V = [v(1), \dots, v(n)]$
 with $A = U \cdot \Sigma \cdot V^T$.
 $u(i)$ called left singular vectors
 $\sigma(i)$ called singular values
 $v(i)$ called right singular vectors
 More generally, if A is $m \times n$ with $m > n$, then
 U $m \times m$ and orthogonal as before
 V $n \times n$ and orthogonal
 Σ is $m \times n$ with same diagonal as before
 When $m > n$, we sometimes write this as follows (the "thin SVD")
 $[u(1), \dots, u(n)] \cdot \text{diag}(\sigma(1), \dots, \sigma(n)) \cdot V^T$
 The same ideas work if A is $m \times n$ with $m < n$.

Geometric interpretation: Thinking of A as a linear mapping from \mathbb{R}^n to \mathbb{R}^m , with the right orthogonal choice of bases of \mathbb{R}^n (i.e. columns of V) and \mathbb{R}^m (i.e. columns of U) then A is diagonal (i.e. Σ):
 $A = U \cdot \Sigma \cdot V^T \Rightarrow A \cdot V = U \cdot \Sigma \Rightarrow A \cdot v(i) = \sigma(i) \cdot u(i)$

Proof: Induction on n :
 Two base cases
 $n=1$: Let U have the first column = $A / \text{norm}(A, 2)$, rest chosen in any way that makes U orthogonal; $\Sigma(1,1) = \text{norm}(A, 2)$, $V = 1$
 $A=0$: Let $U = I_m$, $\Sigma = 0$, $V = I_n$

Induction step (if A nonzero):
 $\|A\|_2 = \max_{\{x: \|x\|_2 = 1\}} \|A \cdot x\|_2$

$$= \max_{\{x: \|x\|_2 = 1\}} \|A*x\|_2$$
 Let $v(1)$ be x attaining the max, $\sigma(1) = \|A\|_2 = \|A*v(1)\|_2$
 and $u(1) = A*v(1) / \|A*v(1)\|_2$.
 Choose $V = [v(1), Vhat]$ to be square & orthogonal
 Choose $U = [u(1), Uhat]$ to be square & orthogonal
 Write $Ahat = U^T * A * V$

$$= \begin{bmatrix} u(1)^T \\ Uhat^T \end{bmatrix} * A * \begin{bmatrix} v(1) \\ Vhat \end{bmatrix}$$

$$= \begin{bmatrix} u(1)^T * A * v(1) & u(1)^T * A * Vhat \\ Uhat^T * A * v(1) & Uhat^T * A * Vhat \end{bmatrix}$$

$$= \begin{bmatrix} \sigma(1) & A12 \\ A21 & A22 \end{bmatrix}$$
 Show $A21 = 0$ by definition of $Uhat$
 Show $A12 = 0$ by definition of $\sigma(1) = \|A\|_2$
 (use part (3) of above Lemma)
 Apply induction to $A22 = U_2 * \Sigma_2 * V_2^T$, so

$$A = U * Ahat * V^T = U * \begin{bmatrix} \sigma(1) & & & \\ & 0 & & \\ & & U_2 * \Sigma_2 * V_2^T & \end{bmatrix} * V^T$$

$$= U * \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} * \begin{bmatrix} \sigma(1) & 0 \\ 0 & \Sigma_2 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & V_2^T \end{bmatrix} * V^T$$

$$= \text{orthogonal} * \text{nonnegative_diagonal} * \text{orthogonal, as desired}$$

The SVD has many useful properties; assume A is $m \times n$ with $m \geq n$.

Fact 1: In the square nonsingular case,

we can use it to solve $A*x=b$ with just $O(n^2)$ more work:

Proof: $X = \text{inv}(A)*b = \text{inv}(U*\Sigma*V^T)*b = (V*\text{inv}(\Sigma)*U^T)*b$

But note: computing the SVD itself is expensive, $O(n^3)$ (see Chap 5)

If all you want to do is solve $A*x=b$, Gaussian Elimination is cheaper. On the other hand, we will see that the SVD is more reliable when A is nearly singular, provides an error bound, and even lets us "solve" $A*x=b$ when A is exactly singular.

Fact 2: When $m > n$, we can solve the full rank least squares problem

$\text{argmin}_x \|A*x-b\|_2$ as follows:

writing the thin SVD, $A = U*\Sigma*V^T$ with U $m \times n$,

then $x = V*\text{inv}(\Sigma)*U^T*b$, same formula as the square case

Proof: Write $A = Uhat*\Sigmahat*V^T$ where $Uhat = [U, U']$ is $m \times m$ and

$\Sigmahat = [\Sigma; 0]$ is $m \times n$. Then

$$\begin{aligned} \|A*x-b\|_2^2 &= \|Uhat*\Sigmahat*V^T*x - b\|_2^2 \\ &= \|Uhat^T*(Uhat*\Sigmahat*V^T*x - b)\|_2^2 \\ &= \left\| \begin{bmatrix} \Sigmahat*V^T*x - U^T*b \\ -U'^T*b \end{bmatrix} \right\|_2^2 \\ &= \left\| \begin{bmatrix} \Sigma*V^T*x - U^T*b \\ -U'^T*b \end{bmatrix} \right\|_2^2 \\ &= \left\| \Sigma*V^T*x - U^T*b \right\|_2^2 + \left\| -U'^T*b \right\|_2^2 \end{aligned}$$

is clearly minimized by choosing $x = V*\text{inv}(\Sigma)*U^T*b$
 to zero out the term depending on x

Def: When $A = U \cdot \text{Sigma} \cdot V^T$ is $m \times n$, $m \geq n$, and full rank, then
 $A^+ = V \cdot \text{inv}(\text{Sigma}) \cdot U^T$ is $n \times m$, and is called the
 Moore–Penrose pseudoinverse of A .

This is the most natural extension of the definition of "inverse" to rectangular full-rank matrices. We can also use the SVD, and an appropriately defined Moore–Penrose pseudoinverse to solve the rank deficient least squares problems, or underdetermined problem ($m < n$), as we describe later. (Q 3.13 has more inverse-like properties of the pseudo-inverse).

Just to solve a least squares problem where you are not worried about rank deficiency, the QR decomposition is cheaper. On the other hand, we will see that the SVD is more reliable when A is nearly singular.

Fact 3: If A symmetric with eigenvalues
 $\text{Lambda} = \text{diag}(\text{lambda}_1, \dots, \text{lambda}_n)$
 and orthonormal eigenvectors $V = [v(1), \dots, v(n)]$,
 then its SVD is
 $A = V \cdot \text{Lambda} \cdot V^T$ (done if all $\text{lambda}_i \geq 0$)
 $= (V \cdot D) \cdot (D \cdot \text{Lambda}) \cdot V^T$ where $D = \text{diag}(\text{sign}(\text{lambda}(i)))$
 $= U \cdot \text{Sigma} \cdot V^T$

Fact 4: Using the thin SVD, the eigenvalue decomposition of
 $A^T \cdot A = (U \cdot \text{Sigma} \cdot V^T)^T \cdot (U \cdot \text{Sigma} \cdot V^T) = V \cdot \text{Sigma}^2 \cdot V^T$

Fact 5: Using the thin SVD, the eigenvalue decomposition of
 $A \cdot A^T = (U \cdot \text{Sigma} \cdot V^T) \cdot (U \cdot \text{Sigma} \cdot V^T)^T = U \cdot \text{Sigma}^2 \cdot U^T$
 (what happens if $m > n$?)

Fact 6: Let $H = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$ be $(m+n) \times (m+n)$, assuming A is $m \times n$
 Then H has eigenvalues $\pm \text{sigma}(i)$
 and eigenvectors $1/\sqrt{2} \cdot [v(i); \pm u(i)]$

Proof: plug in $A = U \cdot \text{Sigma} \cdot V^T$

Fact 6 suggests that algorithms for the SVD and the symmetric eigenproblem will be closely related (see Chap 5)

Fact 7: $\|A\|_2 = \text{sigma}(1)$, $\|\text{inv}(A)\|_2 = 1/\text{sigma}(n)$ and

Def: $\text{kappa}(A) = \text{sigma}(1)/\text{sigma}(n)$ is called condition number of A for reasons we will see shortly

Fact 8: Let S be the unit sphere in \mathbb{R}^n . Then $A \cdot S$ is an ellipsoid centered at the origin with principal axes $\text{sigma}(i) \cdot u(i)$

Proof: suppose $s = [s_1; \dots; s_n]$ where $\sum_i s(i)^2 = 1$, and write

$$A*s = U*Sigma*V^T*s = U*Sigma*shat = \sum_i u(i)*sigma(i)*shat(i)$$

(matlab demo a = randn(2,2), svddemo2; a = randn(3,3); svddemo3)

Fact 9: Suppose

$$sigma(1) \geq \dots \geq sigma(r) > 0 = sigma(r+1) = \dots = sigma(n).$$

Then A has rank r; the null-space of A is

span(v(r+1),...,v(n)), of dimension n-r,

and the range space of A is

span(u(1),...,u(r)), of dimension r

Fact 10: Matrix A_k of rank k closest to A in 2-norm is

$$A_k = \sum_{i=1}^k u_i * sigma(i) * v(i)^T = U * Sigma_k * V^T$$

where $Sigma_k = \text{diag}(sigma(1), \dots, sigma(k), 0, \dots, 0)$

and the distance is $\text{norm}(A - A_k, 2) = sigma(k+1)$

In particular, the distance to the nearest singular

(or non-full rank) matrix is $sigma(n) = sigma_{\min}$.

Proof: easy to see that A_k has right rank, right distance to A;

need to show no closer one:

Suppose B has rank k, so null space has dimension n-k.

The space spanned by $\{v(1), \dots, v(k+1)\}$ has dimension k+1.

Since the sum of the dimensions $(n-k) + (k+1) > n$, these two spaces must overlap (can't have $> n$ independent vectors in R^n).

Let h be unit vector in their intersection. then

$$\begin{aligned} \text{norm}(A-B, 2) &\geq \text{norm}((A-B)*h, 2) = \text{norm}(A*h, 2) \\ &= \text{norm}(U*Sigma*V^T*h, 2) = \text{norm}(Sigma*V^T*h, 2) \\ &= \text{norm}(Sigma*[x(1), \dots, x(k+1), 0, \dots, 0]^T, 2) \\ &\geq sigma(k+1) \end{aligned}$$

(matlab demo: We use this idea that A_k approximates A to demonstrate image compression, since an image is just a matrix (of gray values, say).

```
load clown.mat, [U,S,V]=svd(X);
figure(1), clf, image(X), colormap('gray')
figure(2), clf, for k=[1:10,20:10:200],
    Xk=U(:,1:k)*S(1:k,1:k)*(V(:,1:k))'; ...
    err = norm(X-Xk)/norm(X); compr = k*(200+320+1)/(200*320);
    figure(2), image(Xk), colormap('gray'), ...
    title(['k= ',int2str(k),' err= ', num2str(err),' compression= ',
        num2str(compr)]),
pause, end
```

To see how the error and compression depend on the rank we can plot:

```
figure(3), s = diag(S);
semilogy(1:200,s/s(1),'r',1:200,(1:200)*(200+320+1)/(200*320),'b'),
title('Error in red, compression in blue'), xlabel('rank'), grid
```

(Note: jpeg compression algorithm uses a similar idea, on subimages)

Now we start using this material to analyze the condition number for matrix inversion and solving $Ax=b$: If A (and b) change a little bit, how much can $\text{inv}(A)$ (and $x=\text{inv}(A)*b$) change?

If $|x|<1$, recall that $1/(1-x) = 1+x+x^2+x^3+\dots$
Now generalize to matrices:

Lemma: If operator $\text{norm}(X)<1$, then $I-X$ is nonsingular and
 $\text{inv}(I - X) = \sum_{i=0}^{\infty} X^i$ and $\text{norm}(\text{inv}(I-X)) \leq 1/(1-\text{norm}(X))$

proof: Claim $I + X + X^2 + \dots$ converges:

$\text{norm}(X^i) \leq \text{norm}(X)^i \rightarrow 0$ as i increases

so by equivalence of norms there is some $C>0$ such that

$$\max_{kj} | (X^i)_{kj} | \leq C * \text{norm}(X^i) \leq C * \text{norm}(X)^i$$

and so kj -th entry of $I + X + X^2 \dots$ is dominated by a convergent geometric series and so converges.

$$\text{Claim } (I - X) * (I + X + X^2 + \dots + X^i) \\ = I - X^{i+1} \rightarrow I \text{ as } i \text{ increases}$$

Next $\text{norm}(I + X + X^2 + \dots)$

$$\leq \text{norm}(I) + \text{norm}(X) + \text{norm}(X^2) + \dots \text{ by triangle inequality}$$

$$\leq \text{norm}(I) + \text{norm}(X) + \text{norm}(X)^2 + \dots \text{ by } ||A*B|| \leq ||A|| * ||B||$$

$$= 1 + \text{norm}(X) + \text{norm}(X)^2 + \dots \text{ by def of operator norm}$$

$$= 1/(1 - \text{norm}(X)) \quad \dots \text{geometric sum}$$

(Later: generalize to arbitrary Taylor expansions, like
 $e^X = \sum_i X^i/i!$)

Lemma: Suppose A invertible. Then $A-E$ invertible if

$$\text{norm}(E) < 1/\text{norm}(\text{inv}(A)) \text{ in which case}$$

$$\text{inv}(A-E) = Z + Z(EZ) + Z(EZ)^2 + Z(EZ)^3 + \dots \text{ where } Z=\text{inv}(A) \\ \text{and}$$

$$\text{norm}(\text{inv}(A-E)) \leq \text{norm}(Z)/(1-\text{norm}(E)*\text{norm}(Z))$$

proof: $\text{inv}(A-E) = \text{inv}((I-E*Z)*A) = Z*\text{inv}(I-E*Z)$

exists if $I-E*Z$ invertible i.e. if $\text{norm}(E*Z) < 1$,

i.e. if $\text{norm}(E)*\text{norm}(Z) < 1$ in which case

$$\text{inv}(A-E) = Z*(1+EZ + (EZ)^2 + \dots)$$

Then take norms

What does this say for 1×1 matrices?

Why can't we write $\text{inv}(A-E) = Z + EZ^2 + E^2Z^3 + \dots$?

Finally, we can ask how much $\text{inv}(A)$ and $\text{inv}(A-E)$ differ.

Lemma: Suppose A invertible. If $\text{norm}(E) < 1/\text{norm}(Z)$, then
 $\text{norm}(\text{inv}(A-E)-Z) \leq \text{norm}(Z)^2*\text{norm}(E)/(1-\text{norm}(E)*\text{norm}(Z))$

proof: $\text{inv}(A-E)-Z = Z(EZ) + Z(EZ)^2 + \dots$
 $= ZEZ (I + EZ + (EZ)^2 + \dots)$
 and then take norms

So the relative change in $\text{inv}(A)$ is
 $\text{norm}(\text{inv}(A-E)-Z)/\text{norm}(Z)$
 $\leq [\text{norm}(A)*\text{norm}(Z) * 1/(1 - \text{norm}(E)*\text{norm}(Z))] * [\text{norm}(E)/\text{norm}(A)]$
 $= [\text{norm}(A)*\text{norm}(Z)] * [\text{norm}(E)/\text{norm}(A)] + O(\text{norm}(E)^2)$
 $= \text{condition_number} * \text{relative_change_in_A}$

What does this say for 1x1 matrices?

This justifies the following definition:

Def: condition number $\kappa(A) = \text{norm}(A)*\text{norm}(Z)$

Fact: $\kappa(A) \geq 1$.

proof: $1 = \text{norm}(I) = \text{norm}(A*Z) \leq \text{norm}(A)*\text{norm}(Z)$

Theorem: $\min\{\text{norm}(E)/\text{norm}(A) : A-E \text{ singular}\}$
 $= \text{relative_distance}(A, \{\text{singular matrices}\})$
 $= 1/\kappa(A)$

proof for 2-norm, using SVD:

$\min\{\text{norm}(E) : A-E \text{ singular}\} = \text{sigma_min}(A)$,
 so $\text{relative_distance}(A, \{\text{singular}\}) = \text{sigma_min}(A)/\text{sigma_max}(A)$
 And $\text{norm}(Z) = \text{norm}(\text{inv}(A)) = \text{norm}(V*\text{inv}(\text{Sigma})*U^T)$
 $= \text{norm}(\text{inv}(\text{Sigma})) = 1/\text{sigma_min}(A)$

We've looked at sensitivity of $\text{inv}(A)$, now look at solving $A*x=b$

Now consider $A*x = b$ vs $(A-E)*x' = b+f$ where $x' = x+dx$

subtract to get

$$A*dx - E*x - E*dx = f$$

$$(A-E)dx = f + E*x$$

$$dx = \text{inv}(A-E)*(f + E*x)$$

$$\text{norm}(dx) = \text{norm}(\text{inv}(A-E)*(f + E*x))$$

$$\leq \text{norm}(\text{inv}(A-E))*(\text{norm}(f) + \text{norm}(E)*\text{norm}(x))$$

$$\leq \text{norm}(Z)/(1-\text{norm}(E)*\text{norm}(Z))*(\text{norm}(f) + \text{norm}(E)*\text{norm}(x))$$

$$\text{norm}(dx)/\text{norm}(x)$$

$$\leq \text{norm}(Z)*\text{norm}(A) * 1/(1-\text{norm}(E)*\text{norm}(Z))$$

$$*(\text{norm}(f)/(\text{norm}(A)*\text{norm}(x)) + \text{norm}(E)/\text{norm}(A))$$

$$\leq \text{norm}(Z)*\text{norm}(A) * 1/(1-\text{norm}(E)*\text{norm}(Z))$$

$$*(\text{norm}(f)/\text{norm}(b) + \text{norm}(E)/\text{norm}(A))$$

$$\text{relerr in } x \leq \kappa(A)$$

* something close to 1 unless A nearly singular

* (rel change in b + rel change in A)

Our algorithms will attempt to guarantee that

(*) computed solution of $A*x=b$ is $(A-E)*x' = b+f$ where

$\text{norm}(f)/\text{norm}(b) = O(\text{macheps})$ and $\text{norm}(E)/\text{norm}(A) = O(\text{macheps})$
so relerr in $x \sim O(\kappa(A) \cdot \text{macheps})$

Recall that property (*) is called "backward stability"

Another practical approach: given x' , how accurate a solution is it?
Compute residual $r = A*x' - b = A*x' - A*x = A*(x' - x) = A*\text{error}$
so $\text{error} = \text{inv}(A)*r$ and $\text{norm}(\text{error}) \leq \text{norm}(\text{inv}(A))*\text{norm}(r)$

$\text{norm}(r)$ also determines the backward error in A :

Theorem: The smallest E such that $(A+E)x' = b$ has
 $\text{norm}(E) = \text{norm}(r)/\text{norm}(x')$

proof: $r = A*x' - b = -E*x'$ so $\text{norm}(r) \leq \text{norm}(E)*\text{norm}(x')$
To attain lower bound on $\text{norm}(E)$, choose $E = -r*x'^T/\text{norm}(x')^2$,
in 2 norm.

In other words, if $Ax' - b$ is small, then the backwards error is small,
which is probably the most you should ask for when the entries of A
are uncertain. (Later we will see how to use "iterative refinement",
aka Newton's method, to get a tiny error in x as long as $\kappa(A)$ is
not about $1/\text{macheps}$ or larger, but this is only sometimes justified.)

All our practical error bounds depend on $\text{norm}(\text{inv}(A))$. To actually
compute $\text{inv}(A)$ costs several times as much as just solving $A*x=b$
($2n^3$ versus $(2/3)n^3$) so we will use cheap estimates of $\text{norm}(\text{inv}(A))$
that avoid computing $\text{inv}(A)$ explicitly, and work with small
probability of large error.

The idea is to use the definition

$$\begin{aligned} \|\text{inv}(A)\| &= \max_{\|x\|=1} \|\text{inv}(A)*x\| \\ &= \max_{\|x\|\leq 1} \|\text{inv}(A)*x\| \end{aligned}$$

and do gradient ascent ("go uphill") on $\|\text{inv}(A)*x\|$
on the convex set $\|x\| \leq 1$; one may also start with a random
starting vector. For right choice of norm it is easy to figure out
ascent direction, and each step requires solving $Ax=b$ for some b ,
which only costs $O(n^2)$, (assuming we have already done LU
factorization). In practice it usually takes at most 5 steps or so to
stop ascending so the total costs is $O(n^2)$; see sec 2.4.3 in the
text for details.

In fact there is a theorem (Demmel, Diament, Malajovich, 2000) that
says that estimating $\kappa(A)$ even roughly, but still with some
guarantee (say "to within a factor of 10^6 ", or within any constant
factor) is as about expensive as multiplying matrices, which in turn
is about as expensive as doing Gaussian elimination in the first
place. Since our goal is to spend just an additional $O(n^2)$ to
estimate $\text{norm}(\text{inv}(A))$ given that we have already done Gaussian
Elimination (LU factorization), this theorem implies that we need to
settle for a small probability of getting a large error in our

estimate of $\text{norm}(\text{inv}(A))$.

Where to find implementations of all this?

Matlab: $A \setminus b$ or $[P,L,U]=\text{lu}(A)$ or condest or normest1

LAPACK: xGETRF just for GEPP where $x = S/D/C/Z$

xGESV to solve $A*x=b$

xGESVX for condition estimation, more

xGECON for condition estimation alone

ScaLAPACK: PxGESV , etc