Some selected topics in linear algebra for SSI (2)

Prof. Brooks
Northeastern University
Spring 2011

This talk will relate to parts of
- Chapter 5
- Appendix B
Outline of those Selected Topics

• Some subsurface problems that lead to linear systems of equations
• Some different ways of looking at $n$ equations in $n$ unknowns: in particular, some geometric interpretations
• The geometry of when square linear systems have:
  – exact solutions,
  – no solutions, and/or
  – non-unique solutions
• Problems where you have more equations than unknowns
• Problems where you have more unknowns than equations
• A quick discussion of solving linear systems in Matlab
• The Singular Value Decomposition to analyze linear systems
Outline of these Notes: Linear Systems that are not Square

- What about errors in the model?
- Overdetermined systems: motivation and geometry
- “Solving” overdetermined systems via least-squares
- What about when you can’t make enough measurements?
- Underdetermined systems: motivation and geometry
- A few words on underdetermined systems
- A few words on solving linear systems in Matlab
Reflectance Spectroscopy Again: Errors in our Assumptions

In the reflectance spectroscopy problem, we have several sources of error:

• We may not know, or be able to account for, all the chromophores present, *e.g.*
  
  – some chromophore is in the medium we did not expect, or
  – some chromophore is in the medium that we thought would be negligible but in fact is not

• We may not have exact values for the spectra, *e.g.*
  
  – something in the medium absorbs differently as a function of frequency than we thought it did, or
  – there is an interaction between molecules in the medium that modifies a chromophore’s absorption spectra (*e.g.* it bonded with some molecule in the medium)
  – the experiments we relied on to characterize the spectra are not as accurate as we might have hoped or do not reproduce its behavior in experimental conditions
Reflectance Spectroscopy Again: Errors in our Assumptions

More sources of error:

- There will be error in what we assumed to be known and constant, e.g.:
  - scattering is not estimated correctly, or
  - scattering is not really constant, or
  - the medium is not thick enough that there is no scatter from the more distant boundary, or
  - there is some specular reflection, or
  - there are a small but not really non-negligible fraction of double-scatters
Reflectance Spectroscopy Again: Errors in our Assumptions

Yet more sources of error:

- There will be noise in our measurements: in optics, both
  - electronic (amplifier) noise, usually modeled as Gaussian distributed with equal means (usually zero) and spatially uncorrelated, and
  - shot noise (intrinsic to optical sources), usually modeled as Poisson distributed, meaning the variance is proportional to the mean (so the noise is spatially uncorrelated but not “white” because the means may depend on position)

- So our model of this error for reflectance spectroscopy may include
  1. a signal-independent additive component (electronic noise), and
  2. a signal-dependent, additive component (shot noise).

- Measurement noise can also be multiplicative (say, a gain factor that multiplies the signal) or even more non-linear (say, clipping from saturation).
More on these Errors

- Note that the first 3 error sources on the previous slides are examples of "model error",
- While the last is an example of "measurement error"
- Model error means our model:
  1. does not describe the physics perfectly accurately,
  2. so, does not perfectly predict (noise-free) measurements given true concentrations
- Measurement error means that even if our model were perfect our model equation would not be exact
- Consequence: not really true that
  \[ b = K \cdot c \]
DOT Again: Some Errors in our Assumptions

• Error sources:
  – Diffusion equation is an approximation
  – Linearization is an approximation
  – Estimation of background absorption and scattering are approximate
  – Position of sources/detectors and coupling of light to tissue are approximate
  – Discretization introduces approximations
  – There will be noise in our measurements: since this is optics, both electronic and shot noise
  – ...

• Again, we have both model error and measurement error
One Way to Deal with Error: Use More Equations than Unknowns

Let’s talk about reflectance spectroscopy problem:

• If everything were perfect, there would be no reason to take more measurements than there are unknown chromophores in model.

• In the case of noise and model error, though
  – Solving with the exact number of equations fits
    1. model error, and
    2. noise
      along with the model and the data
  – If we use some “extra” measurements, we can hope to “average” out the error
So Let’s Hedge Our Bets

Example of Using More Equations than Unknowns
Why Might Fitting to More Measurements Make Sense?

The measurement noise often

• has zero average value, so the mean of the noisy measurements is the true (noise-free) measurement

• is uncorrelated with the true measurement

• ... At least that’s often a reasonable assumption ...

So we can

• Hope to average out the + and − errors to decrease their effect.

• Suggests an approximate solution that takes more measurements into account,

• Ideally, the error goes down as $\frac{1}{\sqrt{n}}$ for $n$ measurements.

How about the model error?

• May have components whose error is zero mean

• But much less likely to be uncorrelated with measurement
Rectangular (non-Square) Systems: Overdetermined Case

So we want to consider the case where we have more equations than unknowns:

- $Ax = b$, with $A$
  
  1. having $m$ rows and $n$ columns (we say it is $m$ by $n$ ($m \times n$)) (say, $3 \times 2$)
  
  2. Here $m > n$

Take first two columns of example matrix from Strang:

$$
\begin{bmatrix}
1 & 0 \\
5 & 4 \\
2 & 4
\end{bmatrix}
$$
Geometry of Overdetermined Matrices

For $A \times n, m > n$:

- This is know as an overdetermined (or tall, thin) set of equations
- Since we have only $n$ (2) columns: largest possible dimension of the column space is $n$ (2).
- But the columns are vectors of length $n$ (3).
- So we don’t have enough columns to fill out all of $R^m$ ($R^3$).
- This means that the dimension of the range (or column space) must be $\leq n < m$, since the column space is not “big enough” to fill out $R^m$. 
What are the Implications About Solvability?

- Remember from square systems, two bad things can happen:
  1. Inconsistent measurements (not in the column space)
  2. Invisible solutions (in the null space)
- In the SSI cases we’ve used as motivation:
  - There are possible measurements $b$ that don’t fit our model—$b$ could be any vector in $\mathbb{R}^m$, but our model only can explain those lying in a $n$-dimensional subspace (a plane in 3-space).
  - In fact, if measurements are noisy, the probability is very high that $b$ is not in the column space.
- So in reflectance spectroscopy for example using “extra” wavelengths, we don’t expect to be able to solve the linear equation exactly.
What About the Invisible Solution Components?

• Remember that invisible components of the solution are nullspace vectors, that is non-zero solution vectors \(x\)'s) that produce zero \(b\) (measurements).

• Thus they have the same number of elements as the number of columns, \(n\) (2)

• So is there a vector \(x = [x_1 \ x_2]^T\) such that

\[
\begin{bmatrix}
1 & 0 \\
5 & 4 \\
2 & 4
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\]
Are There Such Invisible Solutions?

- If the columns of $A$ are linearly independent, we can’t find a set of $\{x_i\}$’s that make them 0.
- So there are no non-zero vectors in the nullspace.
- This means we don’t have to worry about “invisible” components in the solution (non-uniqueness).
- In this case:
  - if $b$ happens to be in the column space it is a unique solution.
  - if $b$ is not in the column space there is no (exact) solution.
  - Interpretation: in this last case the problem is “overconstrained”. You don’t have enough degrees of freedom to solve it.
And if the Columns are not Independent . . . ?

- Now, if the columns of $Ax$ are linearly dependent as well as the rows:
  - Then the dimension of the column space is $< n$, which itself is $< m$.
  - So we have some possible concentrations that we can't see in the measurements.
  - Thus in this case:
    * if $b$ is in the column space there are an infinite number of solutions.
    * if $b$ is not in the column space there is no (exact) solution.

- Summary: in the $3 \times 2$ case:
  1. if the columns are independent (as in the figure a few slides back), and if $b$ is in the plane they define, there is a unique solution
  2. if the columns are not independent, and if $b$ is in the line they define, there are infinitely many solutions
  3. if $b$ is not in the column space, whether plane (independent) or line (dependent), there is no exact solution
What Do We Do About Systems with No Exact Solution?

- Consider an overdetermined system of equations:
  \[ A \cdot x = b, \]
  with \( A \) being \( m \times n, m > n \).

- We know from previous discussion that, in general, \( b \) will not be in the range space of \( A \), since we don’t have enough length \( m \) vectors in the \( n < m \) columns of \( A \) to fill out \( R^m \).

- We could choose some set of \( n \) “best” equations and solve it, but
  - Need to choose “best” from \( \binom{m}{n} \) possibilities.
  - Throwing away potentially useful measurements.

- Instead: Let’s find a solution \( x^* \) such that we minimize some measure of the difference between \( A \cdot x^* \) and \( b \).

- Then \( x^* \) will be provide an “optimal fit” to our measurements, in whatever way we define “optimal”.
Least Squares Solutions

- Want a solution that comes “close” to fitting model to measurements.
- How do we measure “close”? How about making the simple differences (errors) between actual measurement and $A \cdot x$,

$$e = b - A \cdot x$$

as small as possible?
- So we need to make the vector $e$ “short”: but how do we measure “length” of a vector?
  - In vectorspeak, length is called a ’norm’.
  - There are many vector norms, but we’ll use the 2-norm:
    * For a vector $x = [x_1, x_2, \ldots, x_n]^T$,

    $$\|x\|_2 = \left[ \sum_{k=1}^{n} x_k^2 \right]^{1/2}.$$

    * Corresponds to usual Euclidean sense of length, by far most commonly used
    * Others include maximum error and using first power (absolute values) instead of second power (squares and square roots)
Why least-squares?

• We minimize the sums of the squared errors, so this is called “least-squares”
• reasonable: intuitive definition of “size” of error
• convenient: taking derivatives leads to linear equations
• optimal: in certain common statistical settings, especially where Gaussian distributions are used.
• related to 'fitting a line to data’, 'regression lines’, etc.
• aside: a computational trick: for a vector $x$
  $$\|x\|_2^2 = x^T x$$
• We will look at solution first geometrically, then algebraically.
The Geometry of Least Squares

Consider visualize-able dimensions: $3 \times 2$ system:

- Red vectors $a_i$ are columns, and plane is column space.
- Blue vector $b$ is right-hand side.
- Pink vectors $Ax_i$ are approximations resulting from using various $x$’s as possible solutions (note they have to be in the column space (the plane))
- Which $Ax_i$ vector in the plane gives the shortest error vector $\|A \cdot x^* - b\|_2$?
Why Be Normal?

Consider any other solution besides the one for which the error vector is orthogonal (normal) to the column space:

- We can decompose that error into a component in the column space plus one orthogonal to the column space.
- By the triangle inequality, the orthogonal error is shorter.
So What Geometry Does This Tell Us?

- For the least-squares optimal solution $x^*$, the error is orthogonal to its approximation to $b$:

$$
(A \cdot x^*)^T (A \cdot x^* - b) = 0.
$$

- Using the fact that $(AB)^T = B^T A^T$, we get

$$
x^{*T} A^T (A \cdot x^* - b) = 0.
$$

- If columns of $A$ are independent, and we want $x^* \neq 0$, then we need to find $x^*$,

$$
A^T A x^* - A^T b = 0,
\text{ or }

A^T A x^* = A^T b.
$$

- Note that the fact of the orthogonality of the error (also called the “residual”) and the projected (onto the column space) solution is enough to specify the solution.

- One example of very powerful and widely used Projection Theorem.
What About Solving This System?

\[ A^T A \cdot x^* = A^T b. \]

- Called the “normal equations”.
- Extremely common and important in many fields: social science, experimental analysis in physical science, statistics, signal processing, communications, . . .
- When do they have a (unique) solution? When the nullspace of \( A^T A \) is empty!
- FACT: the nullspace of \( A^T A \) is the same as the nullspace of \( A \) itself.
- Proof next, but first what does this mean?
  - Recall that \( A \) is \( m \times n \), \( m > n \).
  - From earlier, this means that its nullspace is empty if the \( n \) columns are linearly independent (full column rank).
The nullspace of $A^T A$

- We want to show that it is simply the nullspace of $A$. To do this we need to show that
  1. if a vector $y$ is in the nullspace of $A$ it’s in the nullspace of $A^T A$, and
  2. if a vector $y$ is in the nullspace of $A^T A$ it’s in the nullspace of $A$.

- First, if $y$ is in the nullspace of $A$, then
  \[
  (A^T A)y = A^T (Ay) = A^T 0 = 0,
  \]
  so the first point is satisfied.

- To show the second point, assume $(A^T A)y = 0$.

- Then if we premultiply this by $y^T$, we get
  \[
  y^T A^T A y = 0.
  \]
  But this expression is the 2-norm of the vector $\tilde{y} = Ay$, so if the vector $\tilde{y}$ has length 0 then it must be the zero vector itself.
So What’s the Algorithm?

Assuming we know that $A$ is full column rank, the straight-forward approach is:

- Compute the matrix $A^T A$
- Compute the vector $A^T b$
- Solve the system $A^T A x = A^T b$
- Note that $A^T A$ is a square (and symmetric) matrix so we can solve this system by ordinary methods.
- In good numerical practice, there are ways to do this that avoid explicit computation of $A^T A$.
- Note: in the case that $A$ is square and invertible, this reduces to the original problem $b = Ax$, since
  1. $(AB)^{-1} = B^{-1} A^{-1}$ for any two invertible matrices $A$ and $B$,
  2. Then $(A^T A)^{-1} A^T = A^{-1} A^{-T} A^T = A^{-1}$
- In good numerical practice you can actually solve for the least-squares solution directly from $A$ and $b$ using something called the QR decomposition
Let’s Take Stock of Where We Are

• We have a geometric picture of when matrix-vector problems have a unique solution
• We have a method to solve problems with more equations than unknowns
• Some things still to address
  1. Suppose we can’t get enough equations, that is, we have more unknowns (columns) than measurements (rows)?
  2. Suppose we have enough (or more than enough) equations—at least as many measurements (rows) as unknowns (columns)—but the columns are not linearly independent
  3. Suppose the columns or rows are linearly independent, but just barely . . .
• We’ll deal with the first one of these here
Why Use Less Equations than Unknowns?

This is never a desired situation, however if we are limited by a physical situation as to where we can make measurements:

DOT Reflectance Simulation Geometry: 9 sources, 16 detectors, = 144 measurements, about 10 times as many voxels
More on Less (Equations . . .)

• When we can’t get enough measurements because of:
  1. the shape of the part of the body of interest
  2. need to stay above the surface of the water
  3. not being able to sample the ground very often
  4. the cost/time to acquire data is important
  5. and we want to maintain a certain resolution in our model of the subsurface phenomenon

• then we face this problem:
  Our system of equations has fewer equations than unknowns ($m < n$ if matrix is $m \times n$).

• Problem is underconstrained: called underdetermined.

• Matrix is “short and fat” not “tall and thin”
Summary of Underdetermined Case

- Rows are length $n$ vectors, and we have $m$ of them.
- But $m < n$, so we have “not enough” row vectors to go in all directions of the higher-dimensional row space.
- This means there must be vectors in $\mathbb{R}^n$ which are orthogonal to all the rows.
- And this means that a non-empty nullspace always exists.
- Which in turn (see previous lecture) means that solutions are not unique, since we can add to any solution a second solution in the nullspace and not be able to see it.
- And in fact there are an infinite number of solutions since we can always scale the null-space component of a solution.
- In DOT terms, this would mean that there are different ways the absorption coefficient could be distributed that would look the exactly the same to our measurements, even with no measurement noise.
- Therefore we need some way to choose a unique solution.
The Most Common Solution for Underdetermined Systems

- Most common choice is to pick the *smallest* solution from the infinite number of possible solutions: called **min-norm**.

- Intuitively, this means the null-space component is zero (since otherwise the solution would be longer . . . )

- This criterion leads to the following equation:

\[ x^* = A^T(AA^T)^{-1}b. \]

- Note: in the case that \( A \) is square and invertible, this reduces to the original problem.

- The inverse in this equation exists when \( A \) has rank \( m \) (that is, it is full row rank).

- No particular reason to choose min-norm, and often does not give a good solution !!
So What’s This Algorithm?

Assuming we know that $A$ is full row rank: we want $x^* = A^T(AA^T)^{-1}b$

- Compute the matrix $AA^T$

- Solution 1:
  1. Compute $(AA^T)^{-1}$
  2. Multiply it by $A^T$
  3. Multiply this by the measurement vector

- BUT: Computing matrix inverses is not the best numerical practice
  1. It’s more (computational) work than necessary
  2. It can amplify numerical errors

- So here’s Solution 2:
  1. Solve the system $AA^T u = b$ for $u$.
  2. Multiply $A^T u$ to get $x^*$
Comparison of Least-Squares and Min-Norm Solutions

- Least-squares for overdetermined system—more measurements than unknowns
- Min-norm for underdetermined system—more unknowns than measurements
- Least squares equation is:
  \[ A^T A \cdot x^* = A^T b. \]
- Min-norm is:
  \[ x^* = A^T (A A^T)^{-1} b. \]
- Both reduce to the same equation for square and full-rank \( A \)
- Key remaining questions: what about if system is
  1. Square but not full-rank
  2. Not square and neither full column rank nor full row rank
  3. Most important: Technically full rank but . . . oh so close to not being so

To be continued . . .
Some Comments about Solving Systems in MATLAB

- You can explicitly compute the inverse with the `inv(A)` command.
- However as mentioned earlier this is generally not the best thing to do.
  - In Gaussian elimination you do more work than necessary to get solve a linear system:
  - It’s better to stop the row operations at upper triangular form and use back-substitution.
  - In addition computing full inverse the results are more sensitive to numerical error (accumulation of round-off error, for instance).
- Better to use Matlab command \\.
- This allows Mathworks to use the best current algorithms they can find and you to know nothing about it . . .
Details on the Matlab \ command

• \( x = A \backslash b \)

• Solves square systems.

• For over\text{determined} systems, calculates least-squares solution (using QR).

• For under\text{determined} system, does not find min-norm solution: rather it finds one which minimizes the number of non-zero elements (maximally sparse)

• What about the \((AA^T)^{-1}\) we need for under\text{determined} min-norm solutions? Remember Solution 2 from a couple slides ago:

1. Set \((AA^T)^{-1}b\) equal to some vector \(u\).
2. Then \(u\) is the solution to \((AA^T)u = b\).
3. So solve this system for \(u\), \(e.g.\) by using \(u = AA^T \backslash b\).
4. Then premultiply \(u\) by \(A^T\)
The Best Thing in Linear Algebra: the SVD

It turns out that the single most useful tool in linear algebra for solving systems of equations is a rather non-obvious decomposition of a matrix:

- For any \( m \times n \) matrix \( A \), it turns out that we can write it as the product of three matrices:
  1. An orthogonal matrix \( U \)
  2. A diagonal matrix \( \Sigma \), and
  3. An orthogonal matrix \( V^T \):

\[
A = U\Sigma V^T.
\]

(“orthogonal” will be defined on next slide)

- If \( m \neq n \) we have several options about how to play with the dimensions. We’ll use this one: \( U \) is \( m \times m \), \( \Sigma \) is \( m \times n \), and \( V \) is \( n \times n \).

- This factorization is called the Singular Value Decomposition (SVD), and the (diagonal) entries of \( \Sigma \) are called the “singular values” of \( A \).
Orthogonality of Matrices and Rotation

- **Definition:** Orthogonal matrices: for a square matrix $Q$, if
  \[
  Q^T Q = I
  \]
  then we say $Q$ is orthogonal.

- **Note** this means that $q_i^T q_j = \delta(i - j)$, where
  1. by $q_k$ we mean the $k$'th column of $Q$, and
  2. $\delta(i - j) = 1$ when $i = j$ and 0 otherwise.

- **Property:** the 2-norm of a vector is not changed by multiplying by an orthogonal matrix:
  \[
  \|Qx\|_2^2 = x^T Q^T Q x = \|x\|_2^2.
  \]

- Consider the ellipsoid interpretation of matrix multiplication: since the length does not change, multiplying by an orthogonal matrix corresponds to a pure *rotation*. 
Geometric Interpretation of the SVD

• Remember that multiplying by orthogonal matrices corresponds to a rotation.

• So we can interpret the SVD as a rotation, a scaling, and another rotation:
A Closer Look at $\Sigma$

- Note that definition of the sizes of $U$, $\Sigma$, and $V$ means that

  1. If $A$ is square ($m = n$), $\Sigma$ is square:

     $$
     \Sigma = \begin{bmatrix}
     \sigma_1 & 0 & \ldots & 0 & 0 \\
     0 & \sigma_2 & 0 & \ldots & 0 \\
     \vdots & & \ddots & & \vdots \\
     0 & \ldots & \sigma_i & \ldots & 0 \\
     \vdots & & & & \vdots \\
     0^T & & & & \sigma_m
     \end{bmatrix}
     $$

  2. If $A$ is overdetermined ($m > n$), $\Sigma$ has the form

     $$
     \begin{bmatrix}
     \Sigma_0 \\
     0,
     \end{bmatrix},
     $$

     with $\Sigma_0$ $n \times n$ and $0$ an $(m - n) \times n$ block of zeros

  3. If $A$ is underdetermined ($m < n$), $\Sigma$ has the form

     $$
     [\Sigma_0 \ 0],
     $$

     with $\Sigma_0$ $m \times m$ and $0$ an $m \times (n - m)$ block of zeros
Some Relevant Properties of the SVD

- The SVD many many properties, uses, and interpretations: we’ll concentrate on the ones most relevant here:
  - How to use it to approximately solve singular systems.
  - How to use it to decide if a system is close to singular.
  - How to use it to analyze and get useful results when systems are close to losing full rank (column, row, or both).

- So let’s list a few properties:
  - The entries of \( \Sigma \) are called the “singular values” of \( A \).
  - They are always ordered as \( \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(m,n)} \).
  - The number of non-zero singular values is equal to the rank of \( A \).
  - The rest of the singular values are equal to 0.
  - So if rank is \( r \), we have
    \[
    \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots \sigma_{\min(m,n)} = 0.
    \]
Two More Useful Facts

• FACT 1: Multiplying a matrix by a diagonal matrix:
  1. **Post-multiplying** multiplies each column by the corresponding diagonal element (consider column-oriented multiplication)
  2. **Pre-multiplying** multiplies each row by the corresponding diagonal element (consider dot-product or transpose of post-multiplying).

• FACT 2: A useful identity: if $A$ and $B$ are any matrices and $D$ is a diagonal matrix (assume they’re all $n \times n$ for convenience), then

$$ADB = \sum_{i=1}^{n} d_i a_i b_i^T,$$

where $a_i$ ($b_i$) are the columns of $A$ ($B$), $d_i$ the $i$’th diagonal element.

  – Note that each term in the sum is an $n \times n$ matrix formed by the outer product of a column and a row.
  – Write it down for some simple examples to convince yourself.
  – Each term in the sum is a rank one matrix (each row of the product is a different number times the row vector $b_i^T$).
  – Applies directly to the SVD
Quantifying Close-to-Losing-Rank

• Up to now we have treated linear independence (or full rank) as a binary property . . . you got it or you don’t.

• What about the vectors $[1, 1, 1]^T$ and $[1, 1, 1.00000001]^T$? In principle they’re independent, but in practice they point awfully close to the same direction.

• Since the determinant of a singular matrix is zero, the first thing we might think of to measure ’how close to singular’ is the determinant. However it turns out that
  – The determinant gives the volume of a parallelepiped described by the matrix.
  – Thus if the matrix is very extended in one direction, even if it is very narrow in another, it can have a reasonable determinant but be close to losing rank.

• Much better to use quantity based on the singular values: the “condition number”
The Condition Number

• The condition number $\kappa$ is defined as the $\sigma_1/\sigma_r$: ratio of the largest to the smallest non-zero singular value.

• Geometric interpretation: gives a measure of eccentricity of ellipse.

• Has many other uses (e.g. quantify sensitivity to numerical error)

• Ill-posed problems (as defined in last lecture) have large condition numbers

• Two classes of high-condition-number matrices:
  1. Ones where there is a big jump between 'large' and 'small' SV’s,
  2. Ones where the decay of SV’s is gradual.

• In the former case, we can truncate the smaller ones if we want—gives “low-rank” approximation.

• In the latter case, life is more difficult . . . and more interesting.
A Numerical Example

Consider the matrix

\[
A = \begin{bmatrix}
10^6 & 0 \\
0 & 10^{-6}
\end{bmatrix}
\]

- It's determinant is 1.
- It's condition number is \(10^{12}\).
- Suppose we want to solve \(Ax = b\), for this \(A\) and \(b = [1, 1]^T\). So \(x = [10^{-6}, 10^6]^T\).
- But if \(b = [1, 1.1]^T\), then we get \(x = [10^{-6}, 1.1 \times 10^6]^T\).
- So a change in the measurement by 0.1 causes a change in the solution by \(10^5\)!
- Thus small errors in the 2nd entry of \(b\) will be greatly magnified in any inversion—more on this later.
Evaluating the 'Goodness' of a System of Equations

- In some sense, $\kappa$ defines 'how good', or at least 'how bad', a system of equations is.
  - In the sense of defining 'how independent' or 'dependent' the equations are, or equivalently
  - 'how much new information' each measurement does or doesn't bring.
- No well-defined algorithm to choose a 'good set' of measurements
- But $\kappa$ is a good quantification tool for deciding if a particular set is good, or comparing two sets
- Becomes a tool, for instance, to choose whether a particular set of wavelengths for reflectance spectroscopy is a good set or not
- $\kappa$ close to 1 is perfect (an orthogonal matrix has condition number 1)
- $\kappa$ “large”? $10^3$? $10^6$? $10^9$? Depends partly on the size of the system.
Solving a non-Singular System with the SVD

• First let’s assume $A$ is square ($m \times m$) and full rank (not singular). Then, if $Ax = b$

$$Ax = b$$
$$U\Sigma V^T x = b$$

• Note in this case that all diagonal entries of $\Sigma$ are non-zero and thus its inverse exists and is diagonal with $(i, i)$ entry $1/\sigma_i$.

• Now we premultiply first by $U^T$, then by $\Sigma^{-1}$, then by $V$, using the orthogonality of $U$ and $V$, and finally FACT 2:

$$\Sigma V^T x = U^T b$$
$$V^T x = \Sigma^{-1} U^T b$$
$$x = V \Sigma^{-1} U^T b$$
$$x = \sum_{i=1}^{m} \frac{1}{\sigma_i} v_i u_i^T b.$$
What Does This Do For Us?

- So this gives us another way of solving a linear system. It's not obvious this is very useful.

- But now suppose that $A$ is singular, with rank $r < m$ (so that we have only $r$ non-zero singular values).

- It turns out that a least-squares (min-norm) solution for this rank-deficient case can be achieved if we simply truncate the previous sum at $r$:

$$x^* = \sum_{i=1}^{r} \frac{1}{\sigma_i} v_i u_i^T.$$  

- We can extend this idea to all cases of interest.
Matrix Version of Same Result

• We define a $n \times n$ diagonal matrix $\Sigma^\dagger$:

$$\Sigma = \begin{bmatrix}
\sigma_1 & 0 & \ldots & 0 & 0 \\
0 & \sigma_2 & 0 & \ldots & 0 \\
\ldots \\
0 & \ldots & \sigma_r & 0 \\
0^T & 0
\end{bmatrix}, \quad \Sigma^\dagger = \begin{bmatrix}
1/\sigma_1 & 0 & \ldots & 0 & 0 \\
0 & 1/\sigma_2 & 0 & \ldots & 0 \\
\ldots \\
0 & \ldots & 1/\sigma_r & 0 \\
0^T & 0
\end{bmatrix}$$

• Using FACT 2 in reverse we can write

$$x^* = V\Sigma^\dagger U^T b.$$  

• The matrix multiplying $b$ here is called the “pseudo-inverse” of $A$ and is usually denoted $A^\dagger$.

• Note that when $A$ is invertible, $A^\dagger = A^{-1}$. 
Extending this Approach

We’ve considered a number of situations: assuming we have \( m \) measurements and \( n \) unknowns

- Square non-singular systems: rank = \( m = n \)
- Square singular systems: rank = \( r < m = n \)
- Over-determined, full column rank systems: rank = \( n < m \)
- Over-determined, column rank deficient systems: rank = \( r < n < m \)
- Under-determined, full row rank systems: rank = \( m < n \)
- Under-determined, row rank deficient systems: rank \( r < m < n \)

In all these cases, the SVD approach leads directly to the (min-norm) least-squares solution:

- truncate the sum at \( r \), or equivalently
- define an appropriate \( \Sigma^\dagger \)
Let’s Consider the Full-Rank Overdetermined Case

Can we verify that

\[ x^* = (A^T A)^{-1} A^T b \]

is the same as

\[ V\Sigma^\dagger U^T b? \]

- Starting from \( A = U\Sigma V^T \), with \( \Sigma = [\Sigma_0 \ 0]^T \) and \( \Sigma_0 \) full rank and invertible (since we assumed \( A \) is full column rank).
- Plugging in we get

\[
\begin{align*}
x^* &= (A^T A)^{-1} A^T b \\
&= \left( (U\Sigma V^T)^T U\Sigma V^T \right)^{-1} (U\Sigma V^T)^T b \\
&= \left( V\Sigma U^T U\Sigma V^T \right)^{-1} (V\Sigma U^T)^b \\
&= \left( V\Sigma \Sigma V^T \right)^{-1} (V\Sigma U^T)^b
\end{align*}
\]

where the third line above follows from the order-reversing property of the transpose of a product and the last line from the orthogonality of \( U \).
Continuing on . . .

• Now let’s examine the middle term inside the product of matrices that needs to be inverted:

\[ \Sigma^T \Sigma = [\Sigma_0 \ 0] \begin{bmatrix} \Sigma_0 \\ 0, \end{bmatrix} = \Sigma_0^2, \]

where each diagonal entry of \( \Sigma_0^2 \) is the square of the corresponding singular value.

• So, using this result, the order-reversing property of the inverse of a product, and the orthogonality of \( V \), we get

\[ x^* = \left( V \Sigma_0^{-2} V^T V \Sigma^T U^T \right) b \]

\[ = \left( V \Sigma_0^{-2} \Sigma^T U^T \right) b \]

\[ = V \begin{bmatrix} \Sigma_0^{-1} & 0 \end{bmatrix} U^T b \]

\[ = \sum_{i=1}^{n} \frac{1}{\sigma_i} v_i u_i^T b. \]

• Similar analysis for the other cases on our list . . .
Computing the SVD

- **Always** use someone else’s program . . .

- Seriously, the SVD is not at all straightforward to compute — its utility seems proportional to its non-obviousness.

- Matlab commands
  - \([U,S,V] = \text{svd}(A)\); computes the SVD as we’ve described it.
  - \(S = \text{svd}(A)\); computes just the singular values.
  - \([U,S,V] = \text{svd}(A,0)\); computes the “economy” SVD — just computes the columns of \(U\) and \(V\) that correspond to non-zero singular values.
Some Other Uses of the SVD

Just listing a few here:

• Quantify the gain of a system,

• Analyze approximate inversions for systems with large condition and gradually-decaying singular values,

• Filter signal from noise via subspace approach,

• Build bases for column, row, null, and left null spaces,

• Approximate matrices by other “nearby” matrices

• Close connection to eigenvalue decomposition of $A^T A$ and $AA^T$. 
So Where Does This Leave Us?

- For any kind of system, we can find a least-squares (min-norm) solution, using the SVD:
  1. Square singular systems
  2. Overdetermined full (column) rank systems
  3. Overdetermined (column) rank deficient systems
  4. Underdetermined full (row) rank systems
  5. Underdetermined (row) rank deficient systems

- Unified framework, simply need to take care of zero diagonal entries of $\Sigma$ via truncation.

- What about ill-posed inverse problems, though?