10 Inverse Problems

Part I: Why they may be unstable (and what this means)

In the previous Lecture we have seen that a number of diverse physical problems lead to the same mathematical model. In a simplified form, that model can be written as

$$\frac{dv}{dt} = r(t)v\,, \tag{10.1}$$

where $v(t)$ is the solution and $r(t)$ is the exponential rate of change. In fact, to illustrate the problem that we will study in this Lecture, it will suffice to consider an even simpler model:

$$\frac{du}{dt} = r(t)\,. \tag{10.2}$$

(Verify that (10.2) can be obtained from (10.1) by replacing v with e^u.) In Lecture 9 we considered a situation which, when applied to equation (10.2), is summarized as follows:

Given the rate of change $r(t)$, find the solution $u(t)$.

This is referred to as the ***direct problem*** for Eq. (10.2). In what follows we will refer to the rate $r(t)$ as ***data***.

In this Lecture, we will first explain that the results of Lecture 9 show that this direct problem is ***stable***. That is, any small perturbation to $r(t)$ engenders a small perturbation to the solution $u(t)$. Then we will show that the ***inverse problem*** of determining the data $r(t)$ from the solution $u(t)$ is ***unstable***. That is, certain small perturbations in $u(t)$ may "cause" large perturbations to $r(t)$. Here and below "cause" refers *not* to the physical causality but to the mathematical procedure by which $r(t)$ is *reconstructed* from $u(t)$. This material will be presented in Section 1.

In Section 2 we will show a natural way to minimize (but not to eliminate) the error in $r(t)$ "caused" (in the aforementioned sense) by an error in $u(t)$. In Section 3 we will briefly look at equations that in some sense extend (10.2) and for which the corresponding inverse problems are even more unstable than for (10.2). In Section 4 we will consider an "ultimately" unstable inverse problem, for which arbitrarily small (e.g., within a computer round-off error of 10^{-16} or so) perturbations of the solution can "cause" very large changes in the primordial data.

Finally, in Section 5 we will reformulate the problem of Section 4 as a matrix equation $A\underline{x} = \underline{b}$ and see what difficulties can be encountered when we try to solve it by using the inverse matrix A^{-1} (note the similarity in the name with "inverse problem"!). This will lead us to considering an important characteristic of a matrix called its ***condition number***. In one special case, we will relate the condition number to the eigenvalues of the matrix, and explain what role certain eigenvectors play in the instability of the inverse problem.

To conclude this Introduction, we note that *not all inverse problems* in Nature *are unstable*. However, here and in the next Lecture we will focus only on unstable ones.

10.1 Instability of the inverse problem for (10.2)

At the end of Lecture 9 we pointed out that if $r(t)$ is perturbed by a very fast, but not necessarily small ripple, then the change in $u(t) = \int_0^t r(t_1)dt_1$ is small (specifically, is proportional to the period of the ripple). We will now read this statement backwards:

A small change in

$$u(t) = \int_0^t r(t_1)dt_1$$

may be engendered by a non-small perturbation of $r(t)$.
That is, if we attempt to restore the data $\hat{r}(t)$ from a solution $\hat{u}(t)$ that differs from another solution $u(t)$ by a small ripple, the difference between the corresponding data $\hat{r}(t)$ and $r(t)$ will not, in general, be small. **This is the central theme of this Lecture**.

To add quantitative details to the above general statement, let us estimate the error with which one can derive $r(t)$ from $u(t)$ in the cases when $u(t)$ is smooth and when it has a noisy component. Let us begin with the smooth case. Assume that $u(t)$ has a bounded second derivative, i.e. $|u''(t)| \le 2M$ for some constant M (the origin of factor 2 will transpire soon). The derivative $\frac{du}{dt}$ in (10.2) can be approximated by a forward difference:

$$\frac{du}{dt} \approx \frac{u(t+h)-u(t)}{h}\,, \quad h \ll 1\,. \tag{10.3}$$

What is the error of this approximation? Recall the Taylor series with a remainder for $u(t+h)$:

$$u(t+h) = u(t) + h\cdot u'(t) + \frac{h^2}{2}u''(t^*)\,, \tag{10.4}$$

where $t \le t^* \le t+h$. Substitution of (10.4) into the r.h.s. of (10.3) yields:

$$\begin{aligned}\frac{u(t+h)-u(t)}{h} &= \frac{u(t)+h\cdot u'(t)+\frac{h^2}{2}u''(t^*)-u(t)}{h}\\ &= u'(t)+\frac{h}{2}u''(t^*)\,.\end{aligned}$$

Equivalently,

$$\frac{u(t+h)-u(t)}{h} = r(t) + O(h)\,,$$

where $O(h) \le Mh$ and M was defined above as the bound for $\frac{1}{2}|u''(t^*)|$. For simplicity, we will take this $O(h)$ *to be equal to* Mh; this will not affect our results but will lighten up some formulae. Thus, the previous equation becomes:

$$\frac{u(t+h)-u(t)}{h} = r(t) + Mh\,. \tag{10.5}$$

This equation "says" that when $u(t)$ is sufficiently smooth, one can improve the approximation for $r(t)$ by decreasing the step size h.

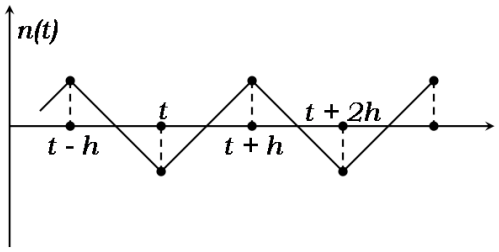
Now consider the other case where a small (of amplitude ε) noisy component is added to $u(t)$, so that now $\hat{u}(t) = u(t) + n(t)$, where $u(t)$ is the same as above. The noisy component $n(t)$ may represent, for example, noise in experimental measurements. Consider now

$$\begin{aligned}\frac{\hat{u}(t+h)-\hat{u}(t)}{h} &= \frac{u(t+h)-u(t)}{h}+\frac{n(t+h)-n(t)}{h}\\ &= r(t)+Mh+\frac{n(t+h)-n(t)}{h}\,.\end{aligned} \tag{10.6}$$

Since $n(t)$ is random, we should identify the worst-case scenario that causes the largest change to the r.h.s. of (10.6) compared to the r.h.s. of (10.5). This occurs when

$$n(t+h) = \varepsilon\,, \qquad n(t) = -\varepsilon\,, \tag{10.7}$$

or vice versa. Graphically, this $n(t)$ is a saw-like ripple shown below:



Then,

$$\frac{n(t+h)-n(t)}{h} = \frac{2\varepsilon}{h}\,,$$

and Eq. (10.6) becomes:

$$\frac{\hat{u}(t+h)-\hat{u}(t)}{h} = r(t) + \left(Mh + \frac{2\varepsilon}{h}\right). \quad (10.8)$$

This formula shows that if we keep taking *smaller and smaller* h to compute the finite difference on the l.h.s of (10.8), the error in the so-obtained approximation for $r(t) = u'(t)$ will eventually *increase* due to the term $\left(\frac{2\varepsilon}{h}\right)$. This means that the problem of reconstruction of the data $r(t)$ from the solution $u(t)$ is ***unstable***: a small (of magnitude ε) noise in $u(t)$ "causes" non-small (of magnitude $\frac{2\varepsilon}{h}$, where $h \ll 1$) noise in the numerically computed $r(t)$. This numerical noise in $r(t)$ *cannot* be eliminated as long as some noise in $u(t)$ is present. However, given the amplitude ε of the noise in $u(t)$, it *is possible to mitigate* the noise in $r(t)$. One way to do so is considered in the next Section. A qualitatively different way will be the subject of the next Lecture.

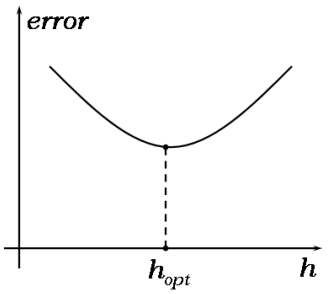
10.2 Minimization of the error in the reconstructed data for Equation (10.2)

For a given solution $u(t)$ (which also determines the constant M in Eq. (10.8)) and a given noise magnitude ε, the error in (10.8) is a function of the only parameter, the step size h of the numerical differentiation:

$$\text{error}(h) = Mh + \frac{2\varepsilon}{h}\,. \quad (10.9)$$

We now use the freedom of choosing h to minimize this expression:

$$(\text{error}(h))' = 0 \qquad \Rightarrow \qquad M - \frac{2\varepsilon}{h^2} = 0 \qquad \Rightarrow \qquad h_{\text{opt}} = \sqrt{\frac{2\varepsilon}{M}}\,.$$



Then the absolute minimum of the error is:

$$\begin{aligned} \text{error}_{\min} &= \text{error}(h_{\text{opt}}) \\ &= M\cdot\sqrt{\frac{2\varepsilon}{M}} + \frac{2\varepsilon}{\sqrt{\frac{2\varepsilon}{M}}} \\ &= 2\sqrt{2\varepsilon M}\,. \end{aligned}$$

Therefore, for

$$h_{\text{opt}} = \sqrt{\frac{2\varepsilon}{M}}\,, \quad (10.10a)$$

one has

$$\hat{r}_{\rm opt}(t) = \left(\frac{\hat{u}(t+h) - \hat{u}(t)}{h} \right)_{\rm opt} = r(t) + O(\varepsilon^{1/2}) \,. \qquad (10.10b)$$

This formula shows that by properly choosing the discretization step h, one can keep the error of the numerically determined data $\hat{r}(t)$ to be relatively small ($O(\varepsilon^{1/2})$), as long as the noise in the available solution is also small ($O(\varepsilon)$).

In a homework problem you will be asked to show that by using a finite-difference approximation to the derivative that is different from the r.h.s. of (10.3), one can slightly improve the error in the numerically obtained $\hat{r}(t)$.

10.3 More unstable inverse problems; the role of smoothness of the solution relative to smoothness of the data

Let us recall the central theme of this Lecture. We are considering situations where a small noisy component in the solution "causes" a non-small noise in the numerically reconstructed derivative $u'(t)$. This can be reworded as follows. Since the operation of *integration reduces* the size of a fast ripple (see the end of Lecture 9), then the inverse operation of *differentiation must amplify* a fast ripple. This is what makes the process of numerical differentiation an unstable problem.

Now, if in some problem, we find the solution by integrating the given data *twice*, the repeated integration will reduce the size of any fast ripple twice (note: *not* by a factor of two, but two times in a row, — one time after the first integration and the other time after the second integration). But if we now want to reconstruct the data from a noisy solution, the repeated differentiation will amplify any fast ripple twice. Thus the instability here will be even stronger than in the inverse problem for Eq. (10.2).

To provide quantitative details to the preceding statement, consider, instead of Eq. (10.2), a problem

$$\frac{d^2\hat{u}}{dt^2} = \hat{q}(t) \,. \qquad (10.11)$$

Its solution $\hat{u}$ is related to the data $\hat{q}$ by:

$$\hat{u}(t) = \int_0^t \left(\int_0^{t_1} \hat{q}(t_2) dt_2 \right) dt_1 \,, \qquad (10.12)$$

where for simplicity we have set the corresponding initial conditions to zero:

$$u_0 = (u')_0 = 0 \,.$$

Let $\hat{q}(t) = q(t) + \tilde{q}(t)$, where $q(t)$ is smooth and $\tilde{q}(t)$ is a *large* but fast ripple:

$$\tilde{q}(t) = A \cdot \cos \omega t, \qquad \omega \gg 1, \quad A \gg 1 \,. \qquad (10.13)$$

We are interested in how the ripple $\tilde{q}$ affects the solution in this case. Since the problem is linear, we have $\hat{u} = u + \tilde{u}$, where u and $\tilde{u}$ are engendered by q and $\tilde{q}$, respectively. Thus, we

consider

$$\begin{aligned}\tilde{u}(t) &= \int_0^t \left(\int_0^{t_1} \tilde{q}(t_2) dt_2 \right) dt_1 \\ &= \int_0^t \left(\int_0^{t_1} A \cos \omega t_2 \; dt_2 \right) dt_1 \\ &= \int_0^t \frac{A}{\omega} \sin \omega t_2 \Big|_0^{t_1} dt_1 \\ &= \frac{A}{\omega^2} (1 - \cos \omega t) \,. \end{aligned} \tag{10.14}$$

Let ω be so large that $\frac{A}{\omega^2} \ll 1$, despite the fact that $A \gg 1$. Then Eq. (10.14) says that having integrated twice a *large* ripple in the data $\hat{q}(t)$, we ended up with a *small* ripple in the solution $\hat{u}(t)$. Therefore, if we try to reconstruct these data by differentiating twice a solution with a small but fast ripple, we should expect a large ripple in the reconstructed data.

We now present a quantitative analysis of such a reconstruction and ripple amplification. We will do so mainly to reiterate the essential steps of a similar analysis in Section 2. It was shown earlier (see Sec. 9.1 of Lecture 9) that a finite difference approximation to $\frac{d^2u}{dx^2}$ is

$$\frac{d^2u}{dx^2} = \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{\Delta x^2} + O(\Delta x^2) \,.$$

For analysis of Eq. (10.11), we replace x with t, Δx with h, and also let $O(h^2) = Mh^2$, similarly to what we did before Eq. (10.5)[11]. Then, rewriting the resulting equation from right to left, we have:

$$\frac{u(t+h) - 2u(t) + u(t-h)}{h^2} = q(t) + Mh^2 \,, \tag{10.15}$$

since $q(t) = u''(t)$. Now let us apply the finite-difference formula on the l.h.s of (10.15) to $\hat{u} = u + n$, where u is smooth, as before, and n is a small noisy component. Then, similarly to (10.6),

$$\frac{\hat{u}(t+h) - 2\hat{u}(t) + \hat{u}(t-h)}{h^2} = q(t) + Mh^2 + \frac{n(t+h) - 2n(t) + n(t-h)}{h^2} \,. \tag{10.16}$$

Repeating the argument made after Eq. (10.6), we look for the "worst" form $n(t)$. Such a worst form must yield the largest value of the numerator on the r.h.s. of (10.16). Inspection reveals that this "worst" $n(t)$ is the same as the one depicted near Eq. (10.8). Since for it, $n(t+h) = \varepsilon \,, \quad n(t) = -\varepsilon \,, \quad n(t-h) = \varepsilon$, Eq. (10.16) becomes:

$$\frac{\hat{u}(t+h) - 2\hat{u}(t) + \hat{u}(t-h)}{h^2} = q(t) + \left(Mh^2 + \frac{4\varepsilon}{h^2} \right) \,. \tag{10.17}$$

As before, we see that decreasing h to zero will eventually make the error grow due to the term $\left(\frac{4\varepsilon}{h^2}\right)$. Then, as in Section 2, we seek a value $h_{\rm opt}$ that would minimize the total error given in the term in parentheses in (10.17). A calculation similar to the one that led to (10.10) yields (verify):

$$h_{\rm opt} = \sqrt[4]{4\varepsilon/M} \,, \tag{10.18a}$$

$$\hat{q}_{\rm opt}(t) = \left(\frac{\hat{u}(t+h) - 2\hat{u}(t) + \hat{u}(t-h)}{h^2} \right)_{\rm opt} = q(t) + O(\varepsilon^{\frac{1}{2}}) \,. \tag{10.18b}$$

[11] However, this "new" M is not related to the "old" M from (10.5).

Thus, optimizing the step size h, we, as before, have been able to make the noise in the numerically reconstructed $\hat{u}(t)$ small. Two notes are now in order.

Note 1 It may appear a little strange that in this problem, which is *more unstable* than the inverse problem for single differentiation, we managed to reduce the ripple to about the same size, $O(\varepsilon^{1/2})$, as for the single differentiation case (see Eq. (10.10b)). However, as you will show in a homework problem, the size of $O(\varepsilon^{1/2})$ for the ripple in the numerically reconstructed $u'(t)$ is *not* the best possible. On the other hand, for the problem of reconstructing $u''(t)$, the ripple size of $O(\varepsilon^{1/2})$ *is* the best possible (details of the corresponding calculation are omitted). Thus, the best possible error in reconstructing $u'(t)$ *is indeed* smaller than that in reconstructing $u''(t)$, as one may intuitively expect.

Note 2 Generalizing the above example to an equation of the form

$$\frac{d^n u}{dt^n} = w(t)\,, \tag{10.19}$$

where one wants to reconstruct the data $w(t)$ from the given solution $u(t)$, we conclude that the greater n, the more unstable this inverse problem is. In other words, the more the direct problem of obtaining u from w smoothens the data w, the more unstable the inverse problem of reconstructing w from u is.

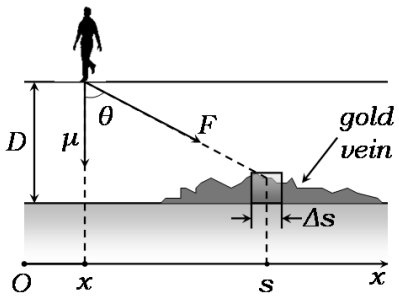
Side note One can show that by choosing an optimal step size $h_{\rm opt}(= O(\varepsilon^{\frac{1}{n+2}}))$, the best possible error in the restored data is $O(\varepsilon^{\frac{2}{n+2}})$, where ε is the magnitude of the noise in the solution.

In the next Section we will consider a problem where an error in the reconstructed data is even larger. We will first derive an equation for the corresponding physical model and then will attempt to solve an inverse problem for it.

10.4 An "ultimately unstable" inverse problem

10.4.1 Derivation of the model

Suppose a geological prospector is searching for a vein of gold; see the posted online Sections 2.3 and 5.3 of the book "Inverse Problems" by C. Groetsch. A more entertaining presentation of the same topic is by the movie "Bluff" with Anthony Quinn and Adriano Celentano (Italy, 1976).



The prospector knows that the vein is located at depth D beneath the ground level, but does not know how gold is distributed in the vein and where exactly the main deposits of gold are. He hopes to find the vein and determine gold distribution in it by measuring a disturbance $\mu(x)$ of the vertical force of gravity at the surface.

A short vein segment having length Δs and located at $x = s$ engenders a force

$$\Delta F(x,s) = G \cdot \frac{M_{\rm meter} \cdot M_{\rm segment}}{r^2} \tag{10.20}$$

on the prospector's force meter. Here G is the gravitational constant, $M_{\rm meter}$ is the meter's own mass, $M_{\rm segment}$ is the mass of the vein segment, and r is the distance between the prospector and the segment. If we denote $w(s)$ to be the (linear) density of gold in the vein, then

$$M_{\rm segment} = w(s) \cdot \Delta s \,.$$

From the Pythagorean theorem and the above figure:

$$r^2 = D^2 + (s-x)^2 \,.$$

For brevity, we denote

$$\gamma \equiv G \cdot M_{\rm meter} \,.$$

Now, the vertical component of this force is (see the figure):

$$\Delta\mu(x,s) = \Delta F(x,s) \cdot \cos\theta = \Delta F(x,s) \cdot \frac{D}{\sqrt{D^2+(s-x)^2}} \,.$$

Finally, the total disturbance in the force (© George Lucas) — the vertical one, that is — equals:

$$\begin{aligned} \mu(x) = \sum_{\text{all } \Delta s} \Delta\mu(x,s) &= \int_{s_{\min}}^{s_{\max}} \frac{\gamma \cdot w(s) ds}{D^2+(s-x)^2} \cdot \frac{D}{\sqrt{D^2+(s-x)^2}} \\ &= \int_{s_{\min}}^{s_{\max}} \frac{\gamma D w(s) ds}{\left(\sqrt{D^2+(s-x)^2}\right)^3} \,, \end{aligned} \tag{10.21}$$

where $s_{\min}$ and $s_{\max}$ are the bounds of the interval where gold is expected to be found. The prospector will collect the measurements $\mu(x)$ at a set of known locations x and will attempt to reconstruct the gold density $w(x)$ from those measurements.

10.4.2 Instability of the inverse problem for Equation (10.21)

We will now demonstrate that a straightforward attempt by the prospector at such a reconstruction is doomed. A reasonable way to approach this reconstruction problem is to rewrite the integral equation (10.21) as a matrix equation. To that end, we assume that the measurements $\mu(x)$ are available at N locations $x_i, i = 1, \ldots N$, and discretize the integral in (10.21) as a finite sum of N terms (see below as to why exactly N). We obtain:

$$\mu(x_i) = \sum_{j=1}^{N} w_j \cdot \frac{c}{\left(\sqrt{D^2+(x_i-x_j)^2}\right)^3} \,; \qquad i,j = 1, \ldots N \,, \tag{10.22}$$

where $w_j = w(x_j)$ and $c = \gamma D \Delta s$. Equation (10.22) is a linear system of N equations for N unknowns w_j, and hence we expect that one should be able to solve it. For future reference, we write it here in the standard general form:

$$A\underline{w} = \underline{\mu} \,, \tag{10.23a}$$

where

$$A_{ij} = \frac{c}{\left(\sqrt{D^2+(x_i-x_j)^2}\right)^3} \,,$$

$$\underline{w} = \begin{pmatrix} w_1 \\ \vdots \\ w_N \end{pmatrix}, \qquad \underline{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix}.$$

We know that this linear system would not have a solution if A were singular, but there is nothing in this model that would suggest that A is singular. Indeed, a singular matrix would result if some of our measurements were linearly dependent on each other, but there is no reason to suspect that this is the case in our problem.

So, the prospector writes a short code where he enters the above matrix A and a known vector of measurements $\underline{\mu}$, and then solves (10.23a) as usual:

$$\underline{w} = A^{-1}\underline{\mu}\,. \tag{10.23b}$$

He gets garbage — a very jagged profile of the gold density $w(x)$ that defies common sense. The prospector panics. He looks for an error in his code, but finds none. Then he thinks the measurements he has collected were insufficient and goes and collects more of them. He augments his matrix A and vector $\underline{\mu}$ with those new entries and repeats the calculation (10.23b). But the new results are even more jagged, and also have nothing in common with the previous ones. What is going on?

To explain the qualitative reason behind this phenomenon, recall what we said in Note 2 at the end of Section 3: The more the direct problem smoothens the data $w(x)$, the more unstable the inverse problem of reconstructing $w(x)$ from $\mu(x)$ is. In the case of Eq. (10.2) with jagged data $r(t)$, the solution $u(t) = \int_0^t r(t_1)dt_1$ is smooth but has a jagged first derivative $r(t)$. This makes the inverse problem for Eq. (10.2) unstable. In the case of Eq. (10.19) with jagged data $w(t)$, the solution $u(t) = \int_0^t dt_1 \int_0^{t_1} dt_2 \ldots \int_0^{t_{n-1}} w(t_n)dt_n$ *and its first* $(n-1)$ *derivatives* are smooth, but the nth derivative (i.e. $w(t)$) is jagged. I.e., for the same jagged data $r(t) = w(t)$, the solution of (10.19) (with $n > 1$) is smoother than the solution of (10.2). This is what made the inverse problem for (10.19) more unstable than the inverse problem for (10.2).

Now let us apply this logic to Eq. (10.21) and look at the derivatives of $\mu(x)$ when $w(x)$ is jagged. $\underline{\underline{All}}$ of the derivatives of $\mu(x)$ are smooth, no matter how jagged $w(x)$ is! Indeed, consider the first derivative (for simplicity we set $s_{\min} = 0$ and $s_{\max} = 1$ in (10.21)):

$$\begin{aligned} \frac{d\mu}{dx} &= \frac{d}{dx}\int_0^1 \gamma D w(s)(D^2 + (x-s)^2)^{-\frac{3}{2}} ds \\ &= \int_0^1 \gamma D w(s) \frac{d}{dx}(D^2 + (x-s)^2)^{-\frac{3}{2}} ds \\ &= \int_0^1 \gamma D w(s) \left[-3(x-s)(D^2 + (x-s)^2)^{-\frac{5}{2}}\right] ds\,. \end{aligned} \tag{10.24}$$

The function in the square brackets in the last line of (10.24) is continuous and bounded (i.e., does not have vertical asymptotes) for all x and s. Therefore, the integral in (10.24) exists for all x, and therefore $\frac{d\mu}{dx}$ is smooth, even if $w(x)$ has jumps! (You will verify this in a homework problem.) The same statement holds for $\frac{d^2\mu}{dx^2}$ and all higher-order derivatives. Thus, the integration in (10.21) has even stronger "smoothing power" than that in (10.19) with a large but finite n. Therefore, the inverse problem for (10.21) should be even more unstable than that for (10.19) (not to mention (10.2)). This is what the prospector discovered the hard way.

10.5 Instability of the inverse problem for Equation (10.21) from the view point of matrix theory

Here we give an explanation of the instability of the inverse problem for the *discretized* equation (10.21), i.e., Eq. (10.22). This will involve the familiar concepts of eigenvectors and eigenvalues of a matrix.

Recall that a singular matrix in Linear Algebra is an analogue of a multiplicative zero for scalars. For example, if A and B are two matrices and A is singular, then $A \cdot B$ is singular. (The same statement holds for scalars $a = 0$ and b and their product $a \cdot b$.) Thus, attempting to solve a matrix equation $A\underline{x} = \underline{b}$ when A is singular is analogous to attempting to solve a scalar equation $0 \cdot x = b$. Now, if one attempts to solve $10^{-6} \cdot x = b$, then $x = 10^6 \cdot b$. As a next logical step, consider two scalar equations:

$$10^{-6} \cdot x = b \qquad \text{and} \qquad 10^{-6} \cdot x = (b + \Delta b)\,, \tag{10.25a}$$

where $\Delta b \ll b$. Obviously,

$$x = 10^6 \cdot b \qquad \text{and} \qquad y = 10^6 \cdot (b + \Delta b)\,,$$

so that

$$(y - x) = 10^6 \cdot \Delta b\,. \tag{10.25b}$$

Thus, even though the difference in the r.h.s of the two equations in (10.25) may be very small (say, $\Delta b = 10^{-3}$), the corresponding solutions x and y will differ by a very large number.

Similarly to how a multiplicative zero is an analogue of a singular matrix, a very small number is an analogue of an *almost* singular matrix. Using this analogy and Eqs. (10.25), one can surmise that two matrix equations with an almost singular matrix A,

$$A\underline{x} = \underline{b} \qquad \text{and} \qquad A\underline{y} = \underline{b} + \underline{\Delta b}\,, \tag{10.26}$$

may have very different solutions $\underline{x}$ and $\underline{y}$, even though the difference $\underline{\Delta b}$ is small. This is indeed the case, at least for a generic $\underline{\Delta b}$. A quantitative measure of how close a matrix is to being singular is called ***the condition number*** of the matrix. The *greater the condition number, the closer to singular a matrix is.* Matrices with very large condition numbers are called ***ill-conditioned***. The definition of the condition number is given in courses on numerical analysis; it is also found in Section 5.1 of C. Groetsch's book. There it is also derived (but you do not have to follow that derivation) that, in the notations of Eq. (10.26),

$$\frac{||\underline{x} - \underline{y}||}{||\underline{x}||} \le \mathrm{cond}(A) \cdot \frac{||\underline{\Delta b}||}{||\underline{b}||}\,. \tag{10.27}$$

Here $||\ldots||$ denotes a norm (e.g., length) of vectors and $\mathrm{cond}(A)$ denotes the condition number of A. Equation (10.27) says that the *relative* error in the r.h.s. vector $\underline{b}$ may be amplified by a factor as large as $\mathrm{cond}(A)$. All major computing software have built-in algorithms for finding the condition number of a matrix; e.g., Matlab's command is simply `cond(A)`.

Below we will derive an expression for the condition number of A in *the special case* where A is symmetric (and real). In this case, as is proved in Linear Algebra, an $N \times N$ matrix A is diagonalizable and has N orthogonal eigenvectors. Since we are considering a nonsingular matrix A, none of its eigenvalues equals 0. Let these eigenvalues be arranged in the increasing order of their absolute values:

$$0 < |\lambda_1| \le |\lambda_2| \le \ldots \le |\lambda_N|\,,$$

and let $\underline{v_1}, \ldots \underline{v_N}$ be the corresponding eigenvectors. Consider Eq. (10.26) where $\underline{b} = \underline{v_N}$ and $\underline{\Delta b} = \varepsilon \cdot \underline{v_1}$, with $\varepsilon \ll 1$. Solving for $\underline{x}$ and $\underline{y}$, one has:

$$\underline{x} = A^{-1}\underline{v_N} \qquad \text{and} \qquad \underline{y} = A^{-1}\underline{v_N} + \varepsilon A^{-1}\underline{v_1}\,. \tag{10.28}$$

Recall that if $(\lambda, \underline{v})$ is an eigenpair of A, then $(\lambda^{-1}, \underline{v})$ is the corresponding eigenpair of A^{-1}. Then:

$$\underline{x} = \frac{1}{\lambda_N}\underline{v_N} \qquad \text{and} \qquad \underline{y} = \frac{1}{\lambda_N}\underline{v_N} + \frac{\varepsilon}{\lambda_1}\underline{v_1}\,,$$

and hence

$$(y - x) = \varepsilon \cdot \frac{1}{\lambda_1}\underline{v_1}\,.$$

Combining the last and first equations above, we have:

$$\frac{||\underline{x} - \underline{y}||}{||\underline{x}||} = \frac{\varepsilon \cdot \frac{1}{|\lambda_1|}||\underline{v_1}||}{\frac{1}{\lambda_N} \cdot ||\underline{v_N}||} = \frac{|\lambda_N|}{|\lambda_1|} \cdot \frac{||\varepsilon \underline{v_1}||}{||\underline{v_N}||} = \frac{|\lambda_N|}{|\lambda_1|} \cdot \frac{||\underline{\Delta b}||}{||\underline{b}||}\,. \tag{10.29}$$

Comparing the r.h.s.'s of (10.27) and (10.29), we can conclude that in *the special case* of a symmetric (and hence diagonalizable) matrix A,

$$\text{cond}(A) = \frac{|\lambda_N|}{|\lambda_1|} = \frac{\max |\lambda| \text{ of } A}{\min |\lambda| \text{ of } A}\,. \tag{10.30}$$

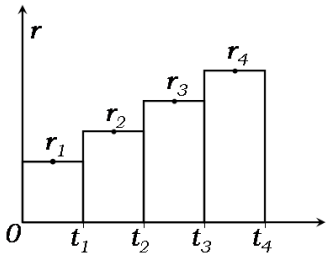
For almost singular matrices, one expects that their minimum eigenvalues are close to zero: $|\lambda_1| \approx 0$, and so $\text{cond}(A) \gg 1$, as stated earlier. In a homework problem you will be asked to illustrate the above derivation geometrically for a simple 2×2 matrix.

To conclude this Lecture, let us look at the previous problems from the viewpoint of the condition number of the matrices involved. Since such a matrix is already available for Eq. (10.21), we comment on it first. The condition number of this matrix increases with N very rapidly, as you will find out in a homework problem. Since that matrix is symmetric, you will also be asked to verify that for it, Eq. (10.30) holds.

Now, to obtain a counterpart of matrix A for Eq. (10.30), we rewrite that equation as an ***integral equation*** (integrating both parts of (10.2) and using the Fundamental Theorem of Calculus):

$$u(t) - u(0) = \int_0^t r(t_1)dt_1\,. \tag{10.31}$$

Let the interval $[0, t_{\max}]$ be subdivided into N subintervals of length $h = \frac{t_{\max}}{N}$ and the integral in (10.31) for each $t = t_j = j \cdot h,\ j = 1, \ldots N$, be computed using midpoint rectangles:



$$\begin{aligned} u_1 - u_0 &= r_1 \cdot h \\ u_2 - u_0 &= (r_1 + r_2) \cdot h \\ u_3 - u_0 &= (r_1 + r_2 + r_3) \cdot h \end{aligned} \tag{10.32}$$

$$\text{etc.}\,,$$

where $u_j = u(t_j)$ and $r_j = r(t_j - \frac{h}{2})$.

Then (10.32) can be written as a linear system:

$$\frac{1}{h}\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \\ 1 & 1 & 1 & 1 & \cdots & 1 \end{pmatrix}\begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_N \end{pmatrix} = \begin{pmatrix} u_1 - u_0 \\ \vdots \\ \vdots \\ \vdots \\ u_N - u_0 \end{pmatrix} \tag{10.33}$$

The matrix in (10.33) is not diagonalizable. (Indeed, all its eigenvalues equal 1, yet it is not the identity matrix.) However, Matlab can still find its condition number. You will be asked to find how it depends on N, in a homework problem.

Finally, we mention a relevant piece of nomenclature. Equations of the form (10.21), which may be written as

$$\mu(x) = \int_a^b K(x,s)w(s)ds\,, \tag{10.34}$$

where $\mu(x)$ is known, a and b do not depend on x, $K(x,s)$ is a continuous function called the kernel, and $w(x)$ is the unknown, are called ***Fredholm equations of the first kind***. As we have seen, the inverse problem for them is very unstable. Equations of the form (10.31) are called ***Volterra equations***; their main difference from (10.34) is that (at least) one limit of integration depends on t. As we have seen, the inverse problem for them is only *mildly* unstable, in the sense that the error in the reconstructed data $r(t)$ can be minimized by a proper choice of h. As a side note, we mention that there are also Fredholm equations of the ***second kind***, which have the form

$$\mu(x) = w(x) + \int_a^b K(x,s)w(s)ds\,. \tag{10.35}$$

The inverse problem for them can be shown (in a course on integral equations) to be stable.