SIO 230 Geophysical Inverse Theory 2009

Supplementary Notes

1. Introduction

In geophysics we are often faced with the following situation. We have measurements made at the surface of the Earth of some quantity, like the magnetic field, or some seismic waveforms, and we want to know some property of the ground under the place where the data were measure. Furthermore, the physics is well understood, and if the property we are seeking were accurately known, we would be able to reconstruct quite accurately the observations that we have taken. Now we wish to infer the unknown property from the measurements. This is the typical *geophysical inverse problem*. It is called an inverse problem, because it reverses the process of predicting the values of the measurements, which is called the *forward problem*. The inverse problem is always more difficult than the forward problem; in fact we have to assume that the forward problem is completely under control before we can even begin to think about the inverse problem, and of course there are plenty of geophysical systems where the forward problem is still incompletely understood, such as in the geodynamo problem, or the problem of earthquake fault dynamics.

Why is the inverse problem more difficult? There are mathematical reasons why recovering unknown functions that appear as factors in differential equations is a complicated business, but the practical issue is less esoteric: the measurements are finite in number and of limited precision, but the unknown property is a function of position, and requires in principle infinitely many parameters to describe it. We are always faced with the problem of *nonuniqueness*: more than one solution can reproduce the data in hand. What can we do?

The most obvious response is to artificially complete the data, by interpolating, filling in the gaps somehow. Then in some circumstances it may be possible to prove a uniqueness theorem, which states that only one model corresponds to each complete data set. When this can be done (which may be hard) you might think the difficulties have been conquered, but that turns out not to be true. Most geophysical inverse problems are *ill-posed* in the sense that they are unstable: then an infinitesimal perturbation of a special kind in the data can result in a finite change in the model. As a consequence the details of the interpolation process used to complete the data are not irrelevant details as one would wish, but they can control gross features of the answer, in contrast to the forward problem, where it is invariably the case that the solution is not only unique, it is stable too.

Another strategy, more commonly used in the past before the advent of larger computers, is to drastically oversimplify the model, for example,

by claiming the unknown structure consists of a small number of layers or zones within which the unknown property is uniform. If there are good geological reasons for doing this, it is still a viable option, but when there is no evidence for this arrangement, even if the simplified model can be made to match the data (and usually it cannot), the inherent nonuniqueness means we are uncertain of the significance of our solution. None-theless geometrical simplification is powerful tool, and often it is the only way to extract useful information. For example, reduction of two- or three-dimensional variations to one dimension may often be a reasonable approximation. The major features of a system can be captured by the assumption that the property varies only with depth, or radius in a spherical Earth, or horizontally.

Clearly if we assume, for example, that electrical conductivity varies only with depth, we are looking for a simple model. The next strategy takes the idea of seeking simplicity explicit while allowing as much complexity as necessary: this is called *regularization*. Here, instead of simplifying the model by reducing its degrees of freedom, we ask for the simplest model consistent with observation. Obviously simplicity can be defined in various ways, but as we will see, the idea is usually to reduce the wiggliness, or roughness in the solution as far as possible. Unstable problems manifest themselves by the introduction of short wavelength oscillations, often of large amplitude, that are not required by the data, but appear because of minor imperfections in the measurements or even because of numerical round-off in the finite-precision computer calculations. By choosing from among the family of models the one with the least roughness, we avoid as far as possible being deceived that there are "interesting" features in the ground, that are in fact accidental. Regularization is today a completely commonplace strategy in inverse theory.

But even after we have obtained the "simplest possible" model by regularization, what do we know with certainty about the Earth? Geophysicists are lamentably quick to assume that the properties of the regularized solution are properties of the true Earth, but that is not guaranteed. If we want to be mathematically rigorous, not a lot is known about the question except for the so-called *linear inverse problems*. For measurements of a single number, we expect to be able to assign an uncertainty, usually an estimate of the standard error derived from statistics: for example, $a = 6371.01 \pm 0.02\,\mathrm{km}$. Why can't we just assign a similar uncertainty to the solution at every point in the model? That seems very reasonable, at first. But, unless we are willing to make other assumptions about the model, assumptions not contained in the measurements, such uncertainties cannot be derived from the data. The reason for this is that it is always possible for a very thin layer to be present, with huge contrasts in value, without making an observable perturbation to the observations. Such a model is therefore consistent with the data, and is in the set of all solutions to the inverse problem. At any point, the allowed deviations can be arbitrarily large and so we cannot (from the measurements

alone) ascribe a point-wise uncertainty.

One solution to this dilemma is to say we are never interested in the model value at a point, only its *average value* over some region. This is a practical matter: in well logs we see large oscillations in properties that we could never expect to match with models based on surface measurements like seismics or magnetics, and so we are always content if the seismic model matches a smoothed version of the well-log record. The uncertainty in a solution may be limited *if we specify an averaging scale*. That is the basis for the *resolution* available in a solution, something we will spend some time on. Averaging over a scale can be useful in its own right and even applies to nonlinear problems; see Medin, Parker, and Constable, 2007. Another idea, hinted at already, is to assume some reasonable model property as an additional *constraint*. For example, if we can plausibly assert that conductivity must increase with depth because of increasing temperature, then that eliminates the very possibility of a thin layer; with this assumption point-wise uncertainties can be computed. See Stark and Parker, 1987. Another popular assumption (but not my favorite) is to assign a probabilistic framework to the problem: in my opinion too much must be assumed (such as Gaussian statistics and a known autocovariance function) without any real justification.

A word on notation. Equations of these Supplementary Notes begin anew at (1) in each numbered section. When I refer to equation outside the current section I will use the form 5(2), which means section 5, equation number (2). When I refer to an equation *Geophysical Inverse Theory* (GIT), I will use the form 2.05(13), which means section 2.05 in Chapter 2, equation (13).

References

Medin, A. E., Parker, R. L., and Constable, S., Making sounding inferences from geomagnetic sounding, *PEPI*, 160, 51-9, 2007.

Parker, R. L., *Geophysical Inverse Theory*, Priceton Univ, Press, 1994.

Stark, P. B. and Parker, R. L., Velocity bounds from statistical estimates of $\tau(p)$ and $X(p)$, *J. Geophys. Res.,* 92, 2713-9, 1987.

2. An Illustration

To give you an idea of the sort of thing we will encounter, here is a seemingly simple, and to some familiar, geophysical problem that has attracted the attention of marine geologists and geophysicists for 40 years. A seamount is a marine volcano, which can be formed at a ridge or in the middle of a plate. Most seamounts are strongly magnetic, and they produce a magnetic anomaly at the sea surface that is easy to observe. A seamount is depicted below in Figure 2.1, and its magnetic anomaly is shown schematically in Figure 2.2. We would like to deduce the internal magnetization vector for the seamount, and in particular the direction of magnetization, because this vector gives paleomagnetic information about the motion of the plate since the time of formation of the volcano – most seamounts form quickly, so the magnetization vector can be used as a kind of snapshot of the paleomagnetic latitude.

Let us first solve the forward problem. The magnetic anomaly is the magnetic field remaining after the main geomagnetic field has been removed, which can be done fairly accurately using satellite models of the longest wavelength fields. The magnetic field due to the seamount is

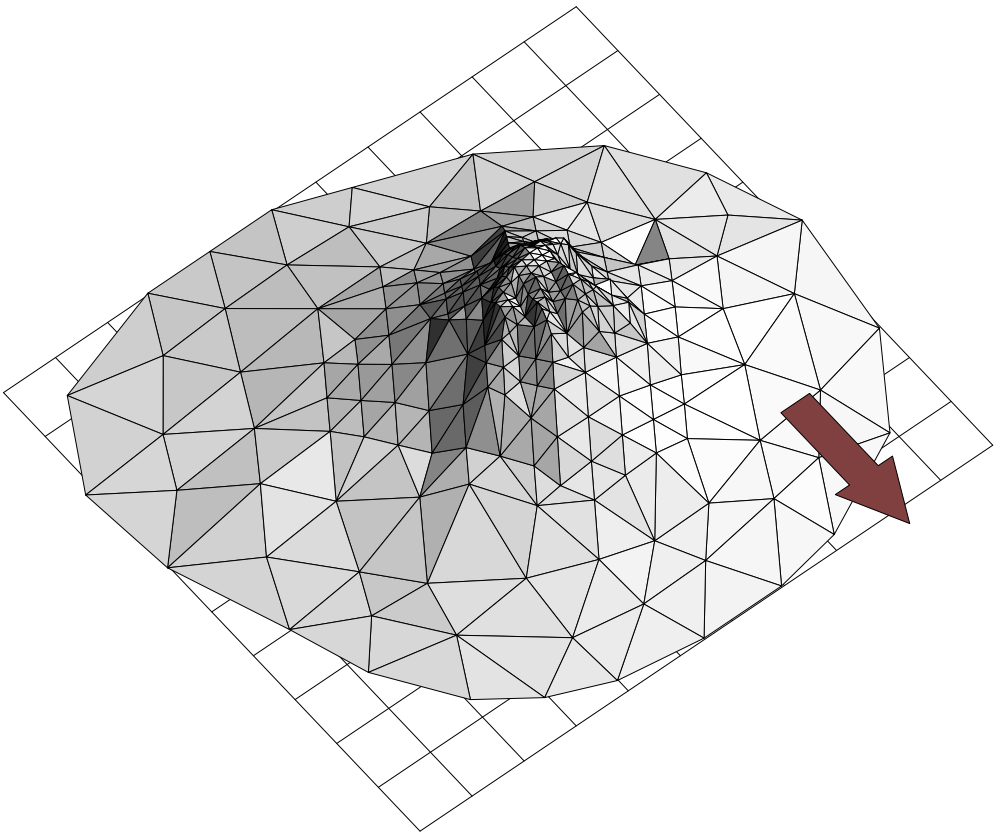


Figure 2.1: Model bathymetry of seamount LR148.8W (48.2°S, 148.8° W). Each square in the base is 5 km on a side and the arrow points north.

given by

$$\Delta\mathbf{B}(\mathbf{r}) = \int_V \mathbf{G}(\mathbf{s}, \mathbf{r}) \cdot \mathbf{M}(\mathbf{s})\, d^3\mathbf{s} \tag{1}$$

where $\mathbf{M}$ is the magnetization vector at the point $\mathbf{s}$ within the seamount V and the vector valued function $\mathbf{G}$ is given by

$$\mathbf{G}(\mathbf{s}, \mathbf{r}) = \frac{\mu_0}{4\pi} \hat{\mathbf{B}}_0 \cdot \nabla\nabla \frac{1}{|\mathbf{r} - \mathbf{s}|} \tag{2}$$

where ∇ acts on $\mathbf{s}$, and $\hat{\mathbf{B}}_0$ is a known constant unit vector and $\mu_0 = 4\pi \times 10^{-7} \mathrm{Hm}^{-1} = 100\,\mathrm{nT\ m\ A}^{-1}$ the permittivity of free space in SI units. All the grad operators here act on the coordinate $\mathbf{s}$. Equations (1)-(2) just state that the observed field at $\mathbf{r}$ is the sum of the fields from all the elementary dipoles within V. If we knew $\mathbf{M}$, which is just the density of dipole moments, we could compute $\Delta\mathbf{B}$ from (1) and (2), which means the forward problem has been solved.

The inverse problem is to discover $\mathbf{M}$ from measurements of $\Delta\mathbf{B}$. Figure 2.2 shows that the actual measurements, which were taken by a Scripps *Thomas Washington* in 1984. Notice that data are collected on an irregular profile. While in reality there are only a few hundred values on $\Delta\mathbf{B}$ on the profile, let us pretend we know $\Delta\mathbf{B}$ everywhere on the ocean surface, and that we know it without error. Surely then we would be in position to determine $\mathbf{M}$. But that turns out to be untrue, because there is

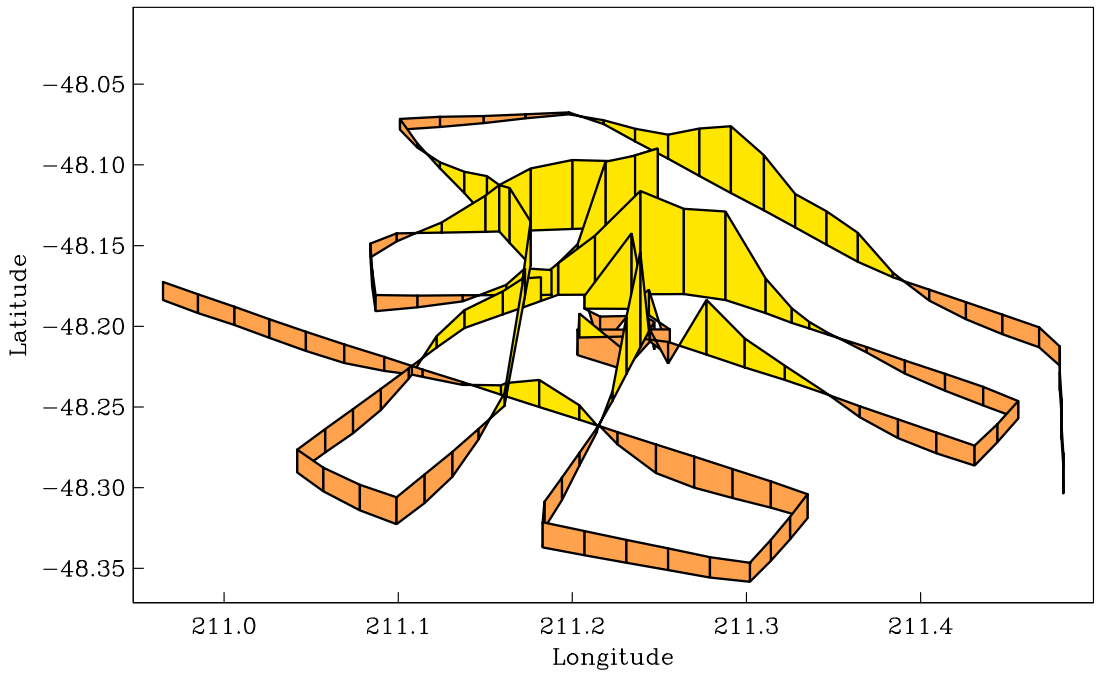


Figure 2.2: Ship track and magnetic anomaly over the seamount LR148.8W.

no uniqueness theorem for this inverse problem, even when perfect data like these are available. We demonstrate the nonuniqueness with some simple vector calculus. First rewrite (1) as

$$\Delta\mathbf{B}(\mathbf{r}) = \int_V \nabla g \cdot \mathbf{M}(\mathbf{s})\, d^3\mathbf{s} \tag{3}$$

where

$$g(\mathbf{s},\mathbf{r}) = \frac{\mu_0}{4\pi}\,\hat{\mathbf{B}}_0 \cdot \nabla \frac{1}{|\,\mathbf{r}-\mathbf{s}\,|} \tag{4}$$

which is OK because $\hat{\mathbf{B}}_0$ is constant. Next consider a magnetization vector inside V given by $\mathbf{m} = \nabla f$ where f is some smooth function. Then (3) becomes

$$\Delta\mathbf{B}(\mathbf{r}) = \int_V \nabla g \cdot \nabla f\; d^3\mathbf{s} \tag{5}$$

$$= \int_V [\nabla \cdot (f\nabla g) - f\,\nabla^2 g]\; d^3\mathbf{s} \tag{6}$$

where I have used the very useful vector identity: $\nabla\cdot(f\mathbf{V}) = \nabla f \cdot \mathbf{V} + f\nabla\cdot\mathbf{V}$. But it is easily seen that $\nabla^2 g = 0$: it is the Laplacian of $1/R$, the potential of a point charge, which vanishes except at the point $\mathbf{r}$; since the observation site is never inside V, it follows that the second term in the integral in (6) vanishes. Next we apply Gauss's Divergence Theorem to the other term:

$$\Delta\mathbf{B}(\mathbf{r}) = \int_V \nabla \cdot (f\nabla g)\; d^3\mathbf{s} = \int_{\partial V} f\hat{\mathbf{n}} \cdot \nabla g\; d^2\mathbf{s} \tag{7}$$

where $\hat{\mathbf{n}}$ is the outward facing normal to the volume V, and ∂V denotes the surface of V. Suppose now I choose any smooth function $f(\mathbf{s})$ that vanishes on ∂V. We see from (7) that the magnetic anomaly $\Delta\mathbf{B}$ due to a magnetization $\mathbf{m} = \nabla f$ *vanishes identically* outside V.

The consequences of this result are that whatever the true magnetization $\mathbf{M}_{\text{true}}$ may be, I can add a magnetization function like $\mathbf{m}$ to it to form

$$\mathbf{M} = \mathbf{M}_{\text{true}} + \mathbf{m} \tag{8}$$

and the new magnetization distribution will match the data just as well as $\mathbf{M}_{\text{true}}$. From (1):

$$\Delta\mathbf{B} = \int_V [\mathbf{G} \cdot [\mathbf{M}_{\text{true}} + \mathbf{m}]\; d^3\mathbf{s} \tag{9}$$

$$= \int_V \mathbf{G} \cdot \mathbf{M}_{\text{true}} d^3\mathbf{s} + \int_V \mathbf{G} \cdot \mathbf{m}\, d^3\mathbf{s} = \int_V \mathbf{G} \cdot \mathbf{M}_{\text{true}} d^3\mathbf{s} + \int_V \mathbf{G} \cdot \nabla f\; d^3\mathbf{s} \tag{10}$$

$$= \int_V \mathbf{G} \cdot \mathbf{M}_{\text{true}} d^3\mathbf{s} + 0 \,. \tag{11}$$

Thus, from the field observations, there is no way to distinguish between the true magnetization and any member of an infinitely large family of alternatives. The magnetization inverse problem does not have a unique solution even with perfect data. The magnetization **m** is called, rather dramatically, a *magnetic annihilator* for this problem.

The first answer to the dilemma of nonuniqueness was the *Drastic Simplification* strategy, introduced for the seamount problem in 1962 (Vacquier, 1962), and still in widespread use today! It is simply asserted that within V the magnetization is uniform, in other words, **M**(**s**) is not a function of position at all, but a constant vector. Then there are exactly three unknowns, the x, y and z components, instead of infinitely many, quite a reduction. In the 1960s there was no compelling evidence to contradict this simple model, but now we know it is wide of the mark. As various seamounts were surveyed magnetically it quickly became clear that the uniform magnetization model was incapable of matching the data, but as I said, people continue to use it to infer paleopoles to this day.

Regularization in this problem (Parker, et al., 1987) takes the following form. We write the magnetization distribution as the sum of two terms:

$$\mathbf{M}(\mathbf{s}) = \mathbf{U} + \mathbf{R}(\mathbf{s}) \tag{12}$$

where **U** is a constant vector, and **R** varies with **s**; obviously any **M** can written this way. To regularize the inverse problem, we ask for the model that makes **R** as small as possible, in other words, we look for the most nearly uniform model that fits the observations. Now we can always match the measurements, and obtain a vector **U** for the uniform part. We have constructed a regularized solution, the kind of thing done all the time in seismic tomography, and surface wave inversion, and magnetotelluric sounding, and on and on. But how reliable is vector **U**, which is the geologically significant product of the calculation? As we will see, in a linear problem like this, **U** is really still completely undetermined, unless we are willing to make some further assumptions, or place additional restrictions on **M**. The information cannot come from observations of $\Delta\mathbf{B}$, which as we have seen by themselves leave a huge amount of ambiguity.

There are several ways we can limit the ambiguity. One approach is to say that we know based on samples of rocks from the seafloor and shallow drill holes in marine basalts that the magnitude of the magnetization $||\mathbf{M}||$ is limited in a way we would be willing to specify. In the Hilbert space machinery that we will soon be studying, the simplest way to do this to introduce a norm of magnetization:

$$||\mathbf{M}|| = \left[\int_V |\mathbf{M}(\mathbf{s})|^2 \, d^3\mathbf{s} \right]^{1/2} . \tag{13}$$

Samples would allow us to say that $||\mathbf{M}|| \leq m_0 V$ with some confidence. We will discuss how to do such calculations later in the class. While this approach is a good idea in principle, it does not work very well for this particular problem in practice. We find the allowed range of directions of $\mathbf{U}$ is very large. And that might be the final answer; after all, it could be that the data we have and the reasonable assumptions we might make, do not in fact allow us to determine $\mathbf{U}$ accurately enough to be geologically interesting.

But we shouldn't give up too soon. A fact of paleomagnetism long exploited in other problems is that rocks are magnetized with a constant direction in large units. It would make no sense for a paleomagnetist to take samples on the surface of an exposed unit if it was not plausible to assume the directions are fairly uniform within. Approximate uniformity of direction is indeed found to be the case, but not for the intensity of magnetization: magnetic intensity is found to vary over two or three orders of magnitude, which is why the oversimplified model of Vacquier doesn't work. So we will restrict the model to be unidirectional in $\mathbf{M}$; we call that direction the unit vector $\hat{\mathbf{M}}_0$. If the geomagnetic field reversed during the formation of the seamount we would need both $+\hat{\mathbf{M}}_0$ and $-\hat{\mathbf{M}}_0$. Most seamounts form rapidly enough that this is a low probability. With that single assumption, unidirectionality, the inverse problem becomes a lot harder to solve, but it turns out, it adds a lot of power to the data and rather good results are obtained; see Parker (1991).

References

Parker, R. L., Shure, L., and Hildebrand, J., The application of inverse theory to seamount magnetism, *Rev. Geophys.,* 25, 17-40. 1987.

Parker, R. L., A theory of ideal bodies for seamount magnetism, *J. Geophys. Res.*, B10, 16101-12, 1991.

Vacquier, V., A machine method for computing the magnetization of a uniformly magnetized body from its shape and a magnetic survey, 123-37, Benedum Earth Magnetism Symposium, Univ. Pittsburgh Press, 1962.

3. Abstract Linear Vector Spaces

This section begins a review of linear algebra and simple optimization problems on finite-dimensional spaces. We will cover some of these problems again but in the more abstract setting of Hilbert space in Chapters and 1 and 2 of GIT. The current segment (Section 3) is a slightly modified version of Section 1.01 in GIT.

The definition of a linear vector space involves two types of object: the **elements** of the space and the **scalars**. Usually the scalars will be the real numbers but occasionally complex scalars will prove useful; we will assume real scalars are intended unless it is specifically stated otherwise. The elements of the space are much more diverse as we shall see in a moment when we give a few examples. First we lay out the rules that define a **real linear vector space** ("real" because the scalars are the real numbers): it is a set $\mathcal{V}$ containing elements which can be related by two operations, addition and scalar multiplication; the operations are written

$$f + g \quad \text{and} \quad \alpha f$$

where $f, g \in \mathcal{V}$ and $\alpha \in \mathbb{R}$. For any $f, g, h \in \mathcal{V}$ and any scalars α and β, the following set of nine relations must be valid:

$$f + g \in \mathcal{V} \tag{1}$$

$$\alpha f \in \mathcal{V} \tag{2}$$

$$f + g = g + f \tag{3}$$

$$f + (g + h) = (f + g) + h \tag{4}$$

$$f + g = f + h, \text{ if and only if } g = h \tag{5}$$

$$\alpha(f + g) = \alpha f + \alpha g \tag{6}$$

$$(\alpha + \beta)f = \alpha f + \beta f \tag{7}$$

$$\alpha(\beta f) = (\alpha\beta)f \tag{8}$$

$$1f = f \ . \tag{9}$$

In (9) we mean that scalar multiplication by the number *one* results in the same element. The notation $-f$ means *minus one* times f and the relation $f - g$ denotes $f + (-g)$. These nine "axioms" are only one characterization; other equivalent definitions are possible. Notice in (7) the meaning of the plus sign is different on the two sides, and in (8) there are two kinds of multiplication going on. An important consequence of these laws (so important, some authors elevate it to axiom status and eliminate one of the others), is that every vector space contains a unique zero element $\mathbf{0}$ with the properties that

$$f + \mathbf{0} = f, \quad f \in \mathcal{V}$$

and whenever

$$\alpha f = \mathbf{0}$$

either $\alpha = 0$ or $f = \mathbf{0}$. If you have not seen it you may like to supply the proof of this assertion.

Here are a few examples of linear vector spaces, most of which we will come across later on. First there is the obvious space $\mathbb{R}^n$. There are two ways to think about this space. One is simply as an ordered n-tuples of real numbers. So an element $\mathbf{x} \in \mathbb{R}^n$ is just

$$\mathbf{x} = (x_1, x_2, \cdots, x_n) \tag{10}$$

where $x_j \in \mathbb{R}$. The definition of addition of two vectors and multiplication by a scalar is self-evident, and if we check off the list of axioms, they are all very obviously true for this collection of elements. The alternative way of looking at $\mathbb{R}^n$ is as the set of column vectors:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{11}$$

This form highlights the fact that $\mathbb{R}^n$ is really a special case of one of the spaces of real matrices, $\mathbb{R}^{m \times n}$; in fact, to use the form (11) we should write the space as $\mathbb{R}^{n \times 1}$.

This brings to the space of real matrices $\mathbb{R}^{m \times n}$. As you could easily guess, it is the collection of all real rectangular arrays of real numbers in the form:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}. \tag{12}$$

Once again it is clear what it means to add two matrices of the same size, and multiplication by a scalar simply means every entry is multiplied by that number. We will be discussing matrix algebra in the next section. Notice that I did not use the proper mathematical notation for this space in GIT, an unfortunate oversight on my part.

Perhaps less familiar to you are spaces whose elements are not finite sets of numbers, but *functions*. For example, consider the real-valued function f that takes a real argument x restricted the closed interval $[a, b]$; a mathematician would write this specification as $f : [a, b] \to \mathbb{R}$, and you should consider it too. The collection of all such functions that are continuous form a space called $C^0[a, b]$. Again, there is no difficulty in seeing what it means to add two such functions, and that the sum of two continuous functions is itself continuous. Scalar multiplication presents

no difficulty either, nor any of the other rules. There is a short table of named function spaces on p 6 of GIT.

In a linear vectors space, you can always add together a collection of elements to form a **linear combination** thus:

$$g = \alpha_1 f_1 + \alpha_2 f_2 + \cdots + \alpha_k f_k \tag{13}$$

where $f_j \in \mathcal{V}$ and the $\alpha_j \in \mathbb{R}$ are scalars; obviously $g \in \mathcal{V}$ too. This kind of operation is at the heart of almost everything we do in linear vector spaces.

To a large extent all that is going on here is *classification*, just a way of organizing things with names we can all agree on. But this is very useful, and you must learn this language and use it.

4. Essential Linear Algebra

The next few lectures will be on linear algebra and its computational aspects. An elementary book for much of the material is by Strang, *Introduction to Applied Mathematics* (Wellesley-Cambridge, 1986); more advanced and a classic in the field is *Matrix Computations*, 3rd Edition, by Golub and Van Loan (Johns Hopkins Univ. Press, 1996) The program MATLAB which you must learn for this class, manipulates matrices very naturally.

A **matrix** is a rectangular array of real (or possibly complex) numbers arranged in sets of m rows with n entries each see 3(12); or equivalently, there are n columns each m long. The set of such m by n matrices is called $\mathbb{R}^{m\times n}$ for real matrices and $\mathbb{C}^{m\times n}$ for complex ones. If $m=1$ the matrix is called a **row vector** and if $n=1$ it is a **column vector**; notice that the space of column vectors will usually be written $\mathbb{R}^m$ rather than $\mathbb{R}^{m\times 1}$. The entries of the array $A\in\mathbb{R}^{m\times n}$ are referred to by indices, a_{ij} which means the entry on the i-th row and the j-th column. A **square** matrix has $m=n$ of course. It is customary to denote a row or column vector by a lower case letter, say x, and then the i-th element is written x_j. Here is a list of names of special matrices defined by the systematic distribution of zeros in them:

Diagonal	$a_{ij}=0$ whenever $i\neq j$
Tridiagonal	$a_{ij}=0$ whenever $\lvert i-j\rvert>1$
Upper triangular	$a_{ij}=0$ whenever $i>j$
Sparse	Most entries zero

Upper triangular matrices are also called *Right triangular*. Lower triangular matrices are defined in the analogous manner, with the inequality reversed. Notice these definitions apply to nonsquare matrices as well as square ones. A diagonal matrix is often conveniently written by specifying its diagonal entries in order, thus:

$$D=\mathrm{diag}(d_1,\,d_2,\,\cdots d_n)\,. \tag{1}$$

In MATLAB this is `D = diag(d)` where `d` is a column or a row vector; MATLAB distinguishes between column and row vectors in most circumstances, so be careful. The square, diagonal matrix with only unity on the diagonal is usually denoted I (a very few authors use E) and is called the **unit matrix**. In MATLAB you form the unit matrix $I\in\mathbb{R}^{m\times n}$ by the weird statement `I = eye(n)`. Sparse matrices are very important because they arise very frequently, and a great of work has been put into numerical algorithms for dealing with them efficiently.

As I mentioned in the previous section, the set of matrices $\mathbb{R}^{m\times n}$ forms a **linear vector space** under the obvious rules of addition:

$$A=B+C \quad \text{means} \quad a_{ij}=b_{ij}+c_{ij} \tag{2}$$

and scalar multiplication:

$$B = c\,A \quad \text{means } b_{ij} = c\,a_{ij} \tag{3}$$

where A, B, $C \in \mathbb{R}^{m\times n}$ and $c \in \mathbb{R}$.

Another basic manipulation, which you all know, is **transposition**:

$$B = A^T \quad \text{means } b_{ij} = a_{ji}\ . \tag{4}$$

Some authors and MATLAB denote transpose by A'; this is to be discouraged in technical writing. Transposition is just the reversal of the roles of the columns and rows. A **symmetric** matrix is its own transpose: $A^T = A$; it is obviously square.

Most important is **matrix multiplication** ($\mathbb{R}^{m\times n} \times \mathbb{R}^{n\times p} \to \mathbb{R}^{m\times p}$):

$$C = A\,B \quad \text{means } c_{ij} = \sum_{k=1}^{n} a_{ik}\,b_{kj}\ . \tag{5}$$

Notice that you can only multiply two matrices when the numbers of columns in the first one equals the number of rows in the second; but the other dimensions are not important, so nonsquare matrices can be multiplied. The combination of the operations of addition and matrix multiplication allows one to define matrix arithmetic analogous to real number arithmetic. The standard arithmetic law of distribution is valid: $A(B+C)=AB+AC$. Less obviously, association of multiplication holds: $A\,(B\,C)=(A\,B)\,C$, when the orders of the matrices are the correct size to permit the product. So we get used to doing algebra on matrices as if they were numbers, **but multiplication is not commutative**. This means in general the order of multiplication matters, that is:

$$A\,B \neq B\,A \tag{6}$$

unless some special property exists.

When one multiplies a matrix into a column vector, there are a number of useful ways of interpreting this operation:

$$y = A\,x\ . \tag{7}$$

If the vectors x and y are in the same space, R^m one can view A as providing a linear mapping or linear transformation of one vector into another. This is especially valuable for 3-vectors and then A represents the components of a tensor (referred to a particular frame). For example, x might be angular velocity about some point, A the inertia tensor, and y would be the angular momentum about the point; or x could be magnetizing magnetic field, A the susceptibility tensor, and y the resultant magnetization vector in a specimen. Another linear transformation performed by matrices in ordinary space is rigid body rotation, used in plate-tectonic reconstruction, and space-ship animation and CAD applications; this involves a special square nonsymmetric matrix to be defined later.

Another useful perspective on matrix multiplication is supplied by thinking about A as the ordered collection of its columns as column vectors:

$$y = A\,x = [a_1,\, a_2,\, \cdots a_n]\,x = x_1 a_1 + x_1 a_2 + \cdots + x_n a_n \tag{8}$$

so that the new vector is just a *linear combination of the column vectors* of A with coefficients given by the elements of x. This is the way we typically think about matrix multiplication when it applies to fitting a model: here y contains data values, the columns of A are the predictions of a theory that includes unknown weight factor given by the entries in x.

We can interpret matrix multiplication this way too. Now let the columns of B be the focus; then

$$A\,B = A\,[b_1,\, b_2,\, \cdots b_p] = [A\,b_1,\, A\,b_2,\, \cdots A\,b_p]\,. \tag{9}$$

In other words, A simply transforms the column vectors of B one at a time. As a matter of fact, it is useful sometimes to partition a large matrix into a set of rectangular submatrices or blocks. Then if you have two matrices, both partitioned into blocks in a consistent manner, you can multiply them together, treating the blocks just as if they were numbers.

There are two ways of multiplying two vectors. The **outer product** if $x \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$:

$$x\,y^T = \begin{bmatrix} x_1 y_1 & \cdots & x_1 y_q \\ . & . & . \\ x_p y_1 & \cdots & x_p y_q \end{bmatrix} \in \mathbb{R}^{p\times q}\,. \tag{10}$$

And the **inner product** of two column vectors of the same length:

$$x^T y = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n = y^T x\,. \tag{11}$$

Of course the inner product is just the vector **dot product** of vector analysis. Notice that if we write A and B as column vectors:

$$A = [a_1,\, a_2,\, \cdots a_m] \text{ and } B = [b_1,\, b_2,\, \cdots b_n] \tag{12}$$

then the matrix product $C = A^T B$ is by definition (5) a collection of inner products:

$$c_{jk} = a_j^T\, b_k,\;\; j = 1, 2 \cdots m,\, k = 1, 2, \cdots n \tag{13}$$

The unit matrix I is special for matrix multiplication. Whenever the product $I\,A$ is permitted, the matrix A is unchanged; the unit matrix plays the role of the number one in arithmetic. Also if A is square and if there is a matrix B so that $A\,B = I$, the matrix B is called the **inverse** of A and is written A^{-1}. When it exists, the inverse matrix is unique. Square matrices A that possess no inverse are called **singular**; when the inverse exists, A is called **nonsingular**. The inverse of the transpose of A is the transpose of the inverse:

$$(A^T)^{-1} = (A^{-1})^T \tag{14}$$

and sometimes authors write $(A^{-1})^T = A^{-T}$. I don't like this notation, however, and so I will never use it.

The idea of the inverse is supposed to be useful for solving linear systems of algebraic equations. When we write (7) and we suppose that the vector y is known and A is square and known too, we can recover the unknown vector x by multiplying both sides of (7) with A^{-1}:

$$A^{-1} y = A^{-1} A\, x = I\, x = x\ . \tag{15}$$

As we will see shortly, calculating the inverse and then multiplying it into y is considered a poor way to do this numerically. How is the inverse actually calculated? We will not go into this in detail, but the ideas behind the numerical calculation of y in (15) are important enough for us to look at later. Once you know how to solve $A\,x = y$, it is a simple step to find A^{-1} from (9) setting $B = I$.

Two cute results. When you transpose the product of two matrices, you can get the same result by transposing first, but you must invert the order:

$$(A\,B)^T = B^T\, A^T\ . \tag{16}$$

And the same goes for the operation of inverting:

$$(A\,B)^{-1} = B^{-1}\, A^{-1} \tag{17}$$

Exercises

4.1 Exhibit an explicit numerical example of a pair of 4 by 4 matrices A and B that commute with each other subject to these rules: neither of them is allowed to be diagonal and A must not be a scalar multiple of B or its inverse. Explain the logical process that led to your answer.

4.2 Our definition, that $A\,B = I$, strictly defines the *right inverse* of A. Prove that if the left inverse exists, it is also B; that is $B\,A = I$; in other words, prove a nonsingular matrix commutes with its inverse. Do not assume the truth of (17). Can you prove a left inverse exists whenever a right one does?

4.3 Show that the inverse of a nonsingular matrix is unique.

4.4 Under what conditions is the product of two symmetric matrices also symmetric?

4.5 Prove (16) and (17).

4.6 In analysis a real self adjoint operator is a linear mapping that satisfies $(x, Ay) = (Ax, y)$ for every vector x, y, where $(\cdot, \cdot)$ is an inner product. Show that for matrices and column vectors, and the inner product (11), that from this definition A must be a symmetric matrix.

Now let us concentrate on square matrices, the kind that arise classically in the solution of linear systems. First, there is the **determinant**.

This is a real number that measures the "volume" of the image of a unit cube after the matrix A has been applied to the space $\mathbb{R}^n$. The value of the determinant isn't much use, except to tell whether it is zero, or not; Here is one way to find it: define $\det(A) = a_{11}$ for $A \in \mathbb{R}^{n\times n}$ with $n = 1$, in other words, a 1×1 matrix. For larger values of n we work up by recurrence:

$$\det(A) = \sum_{j=1}^{n} (-1)^{j+1}\, a_{1j}\, \det(A^{1j}) \tag{18}$$

where A^{1j} is the matrix in $\mathbb{R}^{n-1\times n-1}$ found by deleting row 1 and column j of A. A useful simple case is that of a triangular matrix: the determinant is the product of the diagonal elements. Some other important properties of the determinant which we will not prove:

(a) $\det(AB) = \det(A)\det(B)$
(b) $\det(A^T) = \det(A)$
(c) $\det(A) = 0$, if and only if $A$is singular

The determinant is used for proving theorems but in numerical work it is seldom used.

Here are the definitions of some important kinds of square matrix.

Symmetric	$A^T = A$
Skew-symmetric	$A^T = -A$
Positive definite	$x^T A\, x > 0,\ \ x \neq 0 \in \mathbb{R}^n$
Positive	$a_{ij} > 0$, all i, j
Orthogonal	$A^T A = I$
Normal	$A^T A = A\, A^T$
Projection	$A^T = A$ and $A^2 = A$
Diagonally dominant	$\lvert a_{ii}\rvert > \sum_{j\neq i} \lvert a_{ij}\rvert$

The orthogonal matrix obviously has the properties that

$$Q^{-1} = Q^T \ \text{ and thus } Q\, Q^T = I\ . \tag{19}$$

Consider an orthogonal matrix Q to be composed of column vectors:

$$Q = [q_1, q_2, \cdots q_n]\,. \tag{20}$$

Then the definition and (13) show that the vectors q_j and q_k are always orthogonal when $j \neq k$, and they are of unit Euclidean length. In other words the columns are a collection of mutually orthogonal unit vectors. But (19) shows that this also means the rows have exactly the same property!

As you probably know the orthogonal matrix is the generalization of the operation of rotation and reflection in a mirror in n-dimensional

space. We can show this in several ways. First we can easily show the orthogonal matrix leaves the volume of an element unchanged, because the determinant of Q is ± 1. Here is the proof: From (18) or the idea that I leaves a space unchanged, $\det(I) = 1$. But

$$Q^T Q = I \ . \tag{21}$$

So

$$1 = \det(Q^T Q) = \det(Q)\det(Q^T) = \det(Q)^2 \ . \tag{22}$$

Hence the result. A negative sign indicates the mapping Q involves reflection, as well as rotation. But equal volume transformations aren't necessarily rotations. The key is that the inner product of two any vectors is preserved.

$$(Qx)^T (Q\,y) = x^T Q^T Q\,y = x^T (Q^T Q)\,y = x^T y \ . \tag{23}$$

When $x = y$ this proves the length of every vector is preserved and in addition the angle between any two vectors is unchanged; so any rigid body will be preserved in shape, which is rotation, and possible reflection.

Suppose we are in $\mathbb{R}^3$ and we want to know the orthogonal matrix for a rotation about the origin, on an axis $\hat{\mathbf{n}}$ (a unit vector) and by an angle θ. Then

$$R\,\mathbf{x} = \mathbf{x}\cos\theta + \hat{\mathbf{n}}\,\hat{\mathbf{n}}\cdot\mathbf{x}(1-\cos\theta) + \hat{\mathbf{n}}\times\mathbf{x}\sin\theta \tag{24}$$

where $\times$ is the ordinary vector cross product in $\mathbb{R}^3$. So the elements of R are

$$r_{ij} = \delta_{ij}\cos\theta + (1-\cos\theta)\,n_i\,n_j + \sin\theta \sum_k \varepsilon_{ijk}\,n_j \tag{25}$$

where ε_{ijk} is the **alternator** with the properties: $\varepsilon_{ijk} = 0$, whenever two indices are equal; $\varepsilon_{ijk} = +1$ whenever ijk is a cyclic permutation of 123; $\varepsilon_{ijk} = -1$ otherwise. You may find this formula useful one day.

Here is another interesting orthogonal matrix – the elementary **Householder matrix**. Suppose $u \in \mathbb{R}^n$ is of unit Euclidean length, meaning $u^T u = 1$. Then the matrix

$$Q = I - 2u\,u^T \tag{26}$$

is orthogonal. Proof: First note that Q is symmetric; therefore

$$Q^T Q = Q^2 = (I - 2u\,u^T)(I - 2u\,u^T) \tag{27}$$

$$= I - 4u\,u^T + 4u\,u^T u\,u^T = I \text{ [QED]} \ . \tag{28}$$

So Q is both symmetric and orthogonal. (Can you think of a general specification of the class of symmetric orthogonal matrices?) The transformation of Q is quite simple to visualize as follows: Consider the vector $y = Q\,x$ $= x - 2u\,(u^T x)$. The vector $u\,(u^T x)$ is the component of x lying in the u direction. So y has that component reversed. That means x has been

reflected in the subspace normal to the direction n. (This means $\det(Q) = -1$; can you prove this independently?) The reason these particular matrices are important is that they play a central role in a special matrix factorization called **QR**, invented by Alston Householder in the 1950s, and the QR method solves least-squares problems; more of this later.

Next we go over some further ideas about linear vector spaces. First, a set of vectors $\{a_1, a_2, \cdots a_n\}$ in a linear vector space (for example, $\mathbb{R}^m$) is said to be **linearly independent** if

$$\sum_{j=1}^{n} \beta_j a_j = 0 \text{ implies } \beta_1 = \beta_2 = \cdots = \beta_n = 0 \tag{29}$$

where $\beta_j \in \mathbb{R}$. If a nontrivial linear combination of vectors can equal the zero vector of the space, the set is called **linearly dependent**. A **subspace** of linear vector space is a set of vectors that is also a linear vector space. For example, any plane in ordinary space through the origin of coordinates; but not a plane that misses the origin. The **spanning set** or more simply the **span** of a collection of vectors, is the linear vector space that can be built from that collection by taking linear combinations of them. Formally,

$$\text{span}\{a_1, a_2, \cdots a_n\} = \{\sum_j \beta_j a_j \mid \beta_1, \beta_2, \cdots \beta_n \in \mathbb{R}\} . \tag{30}$$

A linearly independent spanning set forms a **basis** for a vector space. The number of elements in a basis is called the **dimension** of the space, and it can be proved every basis for a particular space has exactly the same dimension, which is a number that characterizes the "size" of the vector space. In intuitive terms, the dimension is the number of free parameters needed to specify uniquely an element in the space. Every collection of more than n vectors in an n-dimensional space must be linear dependent. These ideas apply quite generally to linear vector spaces (that might contain functions or operators), but we are interested here on elements that are collections of real numbers.

For a matrix in $\mathbb{R}^{m\times n}$ there are two important spaces. First the **range space**, also called the **column space** of A, which we write $\mathcal{R}(A)$. It is simply the linear vector space formed by taking linear combinations of the column vectors of A. Recall (8); this means that for all $x \in \mathbb{R}^n$

$$Ax \in \mathcal{R}(A) . \tag{31}$$

Obviously

$$\text{If } A = [a_1, a_2, \cdots a_n] \text{ then } \mathcal{R}(A) = \text{span}\{a_1, a_2, \cdots a_n\} . . \tag{32}$$

The dimension of the column space of a matrix in $\mathbb{R}^{m\times n}$ can never be more than n the number of columns but it can be less. That dimension is so important it has its own name: the **rank** of A:

$$\text{rank}(A) = \dim[\mathcal{R}(A)] . \tag{33}$$

It can be shown that $\text{rank}(A) = \text{rank}(A^T)$, so the rank of a matrix is the maximal number of linearly independent rows or columns. A matrix in $\mathbb{R}^{m \times n}$ is said to be of **full rank** if $\text{rank}(A) = \min(m, n)$ and to be **rank deficient** otherwise.

The other side of the coin of the columns space is the **null space** of A. This is given by the set of xs that cause $Ax = 0$: for $A \in \mathbb{R}^{m \times n}$

$$\mathcal{N}(A) = \{x \in \mathbb{R}^n \mid Ax = 0\} \tag{34}$$

We agree to define as zero, the dimension of the space comprising a single element, the zero vector, then

$$\dim[\mathcal{N}(A)] = n - \text{rank}(A). \tag{35}$$

For the important case $m = n$ the following are equivalent:

(a) A is nonsingular
(b) $\dim \mathcal{N}(A) = 0$
(c) $\text{rank}(A) = n$
(d) $\det(A) \neq 0$

References

Golub, G. H., and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1983.

Strang, G., *Introduction to Applied Mathematics* Wellesley-Cambridge Press, 1986

5. Simple Least Squares Problems

Least squares problems are examples of optimization problems that involve the simplest of **norms**. We are going to solve these problems in several ways, to illustrate a the use of use of Lagrange multipliers and a few other things. The bible for the numerical aspects of LS is the ancient Lawson and Hanson, 1974. First what is a norm? Informally, a norm is a real number that measures the size of an element in a linear vector space. Assigning a norm to a linear vector space is said to **equip** the space with a norm. Most linear vector spaces can be so equipped (spaces of functions, operator, matrices, etc), but here we will consider only the simplest norm for $\mathbb{R}^m$, the Euclidean length: when $x \in \mathbb{R}^m$

$$|| x || = \sqrt{\sum_{k=1}^{n} x_k^2}\,. \tag{1}$$

We can obviously express this in other ways, for example $|| x || = (x^T x)^{1/2}$. The space $\mathbb{R}^m$ is then properly called E^m, but we won't be pedantic. Here is an approximation problem often encountered in geophysics, the classical least squares problem. We will state it as a problem in linear algebra.

Suppose we are given a collection of n vectors $a_k \in \mathbb{R}^m$ and we wish to approximate a target vector y by forming a linear combination of the a_k; when $n < m$, as we shall assume, we will not expect to be able to do this exactly, and so there will be an error, called in statistics the **residual**:

$$r = y - \sum_{k=1}^{n} x_k a_k\,. \tag{2}$$

In data analysis, straight-line regression is in this form, or fitting any simple linear model to a data set. In numerical analysis you might want to approximate a complicated function by a polynomial. To get the *best approximation* in some sense, we want the size of the vector $r \in \mathbb{R}^m$ to be as *small* as possible. Once we've picked a way to measure the size, we have a minimization problem. The simplest norm for computational purposes is the Euclidean length, and this leads to the **overdetermined least squares problem**. If we can rewrite (2) in matrix notation:

$$r = y - Ax \tag{3}$$

where $x \in \mathbb{R}^n$ and the matrix $A \in \mathbb{R}^{m \times n}$ is built from columns that are the a_k:

$$A = [a_1, a_2, \cdots a_n]\,. \tag{4}$$

So the minimization problem is to solve

$$\min_{x \in \mathbb{R}^n} \; f(x) \tag{5}$$

where

$$f(x) = || r ||^2 = r^T r = (y - Ax)^T (y - Ax)\,. \tag{6}$$

Obviously we can square the norm if it simplifies the algebra.

I will offer you several solutions to this problem, some of which may be unfamiliar. First the classical approach, which is to multiply descend into subscripts, and differentiate:

$$f = \sum_{j=1}^{m} r_j^2 . \tag{7}$$

Then

$$\frac{\partial f}{\partial x_k} = 2 \sum_{j=1}^{m} r_j \frac{\partial r_j}{\partial x_k} \tag{8}$$

$$= 2 \sum_{j=1}^{m} (y_j - \sum_{i=1}^{n} a_{ji} x_i) \times (-a_{jk}) \tag{9}$$

$$= -2 \sum_{j=1}^{n} a_{jk} y_j + 2 \sum_{i=1}^{n} (\sum_{j=1}^{m} a_{jk} a_{ji}) x_i \tag{10}$$

and this is true for each value of k. At the minimum we set all the derivatives to zero, which leads to:

$$\sum_{i=1}^{n} (\sum_{j=1}^{m} a_{jk} a_{ji}) x_i = \sum_{j=1}^{n} a_{jk} y_j, \quad k = 1, 2, \cdots n . \tag{11}$$

Translated into matrix language these are the so-called **normal equations**:

$$A^T A \, x = A^T y . \tag{12}$$

Note that $A^T A$ is a square n by n matrix, and the left side is a column n-vector. So the unknown expansion coefficients are found by solving this system of linear equations, formally by writing

$$x = (A^T A)^{-1} A^T y \tag{13}$$

a result that should be familiar to you.

This answer looks ugly and seems to have no intuitive content. But a geometrical interpretation can help a lot. Suppose we assume that the vectors a_k are linearly independent (which they must be if we can write (13)). Then the collection of all vectors that can be formed from linear combinations of them is a **subspace** of $\mathbb{R}^m$ which we will call $\mathcal{A}$; it is the column, or range, space of the matrix A, and so $\mathcal{A} = \mathcal{R}(A)$ from 4(33). The approximation problem we are solving can be stated as finding the vector in $\mathcal{A}$ that comes as close to y as possible. We rewrite (12) as

$$0 = A^T (A \, x - y) = A^T r \tag{14}$$

$$= \begin{bmatrix} a_1^T r \\ a_2^T r \\ \vdots \\ a_n^T r \end{bmatrix}. \tag{15}$$

Remember the zero on the left is the vector $0 \in \mathbb{R}^n$. So what this equation is saying is that the residual vector, the error in the approximation to y, is orthogonal to every one of the basis vectors of the space $\mathcal{A}$ (because $a_1^T r$ is the dot product between a_1 and r). And that is what you might expect from a geometrical interpretation as shown in Figure 5.1.

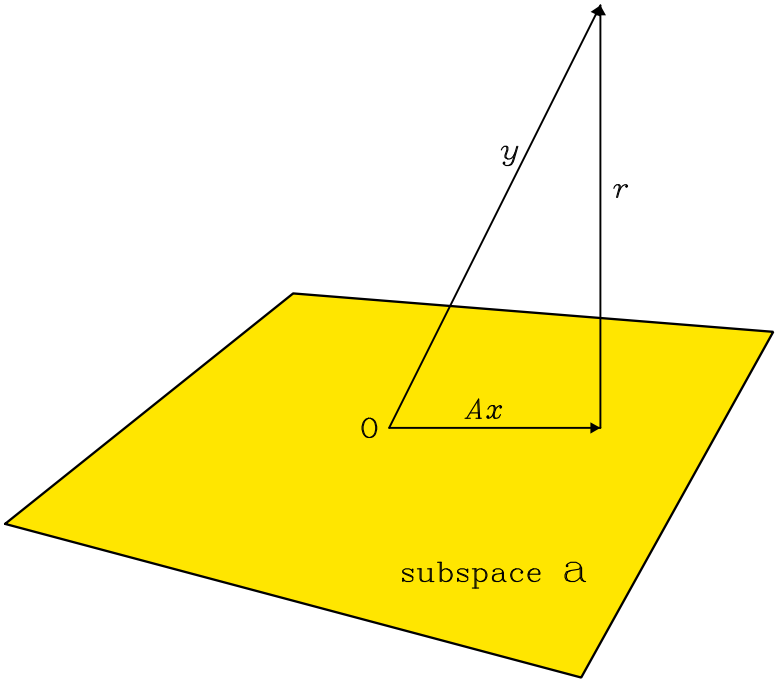
Let us give a name to the approximation we have created, let $Ax = \tilde{y}$. Then $\tilde{y}$ is called the **orthogonal projection** of y into the subspace $\mathcal{A}$. The idea of a projection relies on the **Projection Theorem** for Hilbert spaces. The theorem says, that given a subspace like $\mathcal{A}$, every vector can be written uniquely as the sum of two parts, one part that lies in $\mathcal{A}$ and a second part orthogonal to the first. The part lying in $\mathcal{A}$ is orthogonal projection of the vector onto $\mathcal{A}$. Here we have

$$y = \tilde{y} + r\,. \tag{16}$$

There is a linear operator, $P_{\mathcal{A}}$ the projection matrix, that acts on y to generate $\tilde{y}$, and we can see that

$$P_{\mathcal{A}} = A\,(A^T A)^{-1} A^T\,. \tag{17}$$

Figure 5.1: Orthogonal projection of y onto the column space of A.



Recall from Section 4 that a projection matrix must be symmetric and satisfy $P^2 = P$. The second property is natural for a projection, because acting once creates a vector falling into the given subspace, acting again leaves it there. Verify these properties for $P_{\mathcal{A}}$.

We describe next a completely different way of looking at the least-squares (LS) problem, which often offers considerable improvement in numerical accuracy. Let us return to Householder's **QR factorization** of a matrix, mentioned briefly in the previous section: every matrix $A \in \mathbb{R}^{m\times n}$ where A is tall (meaning $m \geq n$) can be written as the product:

$$A = Q\,R \tag{18}$$

where $Q \in \mathbb{R}^{m\times m}$, $R \in \mathbb{R}^{m\times n}$, and Q is *orthogonal*, and R is *upper triangular*; recall that upper triangular means all zeros below the diagonal so that $R_{ij} = 0$ when $i > j$. We can write

$$R = \begin{bmatrix} R_1 \\ O \end{bmatrix} \tag{19}$$

where $R_1 \in \mathbb{R}^{n\times n}$ and is also upper triangular. For how the QR factorization is found in practice and why QR is numerically stable, see GIT 1.13 and the references there. To solve the LS problem we look to the Euclidean norm of r:

$$\|\,r\,\| = \|\,y - Ax\,\| = \|\,y - QRx\,\| . \tag{20}$$

Recall that $Q\,Q^T = I$, so

$$\|\,r\,\| = \|\,QQ^T y - QRx\,\| = \|\,Q(Q^T y - Rx)\,\| . \tag{21}$$

Now recall that the length of a vector is unchanged under mapping with an orthogonal matrix: $\|\,z\,\| = \|\,Qz\,\|$. So

$$\|\,r\,\| = \|\,Q^T y - Rx\,\| = \|\,\hat{y} - Rx\,\| . \tag{22}$$

Next square the norm and partition the arrays inside the norm into two parts, the top one with n rows:

$$\|\,r\,\|^2 = \left\|\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} - \begin{bmatrix} R_1 \\ O \end{bmatrix} x \right\|^2 = \left\|\begin{bmatrix} \hat{y}_1 - R_1 x \\ \hat{y}_2 \end{bmatrix}\right\|^2 \tag{23}$$

$$= \|\,\hat{y}_1 - R_1 x\,\|^2 + \|\,\hat{y}_2\,\|^2 . \tag{24}$$

The second term $\|\,\hat{y}_2\,\|^2$ in the sum is indifferent to the choice of x; but we can reduce the first term to zero by solving

$$R_1 x = \hat{y}_1 . \tag{25}$$

So this must be the solution to finding the smallest norm of r. Because R_1 is upper triangular, (25) is solved by **back substitution**, starting at the bottom and working upwards, which is very simple. This doesn't look like a very efficient way to find the LS answer, but it can be made very

efficient: for example, there is no need to store the matrix Q, because one can calculate the vector $\hat{y} = Q^T y$ without it. The QR factorization is competitive with the normal equations for execution times (it is slightly slower but, as mentioned earlier, it is numerically more stable against the accumulation of numerical error. The key to understanding the accuracy in the solution of $Ax = b$ is $\kappa(A) = \| A \| \ \| A^{-1} \|$ called the **condition number** of A, which estimates the factor by which small errors in b or A are magnified in the solution x. This can sometimes be very large ($> 10^{10}$). It is shown in GIT that the condition number in solving the normal equations (13) is *the square* of the condition number for (25), which can sometimes lead to catastrophic error build up. Therefore, for not too large systems, QR is the proper way to go. In MATLAB, while you can get the QR factors with the call

```
[Q R] = qr(A);
```

the LS problem is *solved automatically* for you by the QR method if you simply write

```
x = A\y;
```

Finally, suppose you substitute the QR factors into the expression for the projection matrix. We find after some algebra that

$$P_{\mathcal{A}} = Q^T \begin{bmatrix} I_n & 0 \\ 0 & O \end{bmatrix} Q \tag{26}$$

where $I_n \in \mathbb{R}^{n \times n}$ is the unit matrix and the rest of the entries are zero. Numerically this way of finding the projection is very stable because one never needs to solve a linear system. But (26) also shows that if one imagines rotating the data space onto new coordinates with Q, the projection operator then becomes the matrix in the middle of (26), which is the projection that simply zeros out all the components of a vector after the nth one.

Exercises

5.1 Show that the last $m - n$ columns in the factor Q of the QR factorization are never used in the LS calculation.

5.2 The *Gram-Schmidt* process is a method of creating a set of orthonormal vectors from a given ordered set of linearly independent vectors by forming linear combinations of one, then two, then three, etc, of the given vectors. Show how the QR process does the same thing.
Hint: First show that the inverse of an right triangular matrix is also right triangular.

Reference

Lawson, C. L., and Hanson, R. J., *Solving Least Squares Problems,* Prentice-Hall, Englewood Cliffs, NJ, 1974.

6. Lagrange Multipliers and More Least Squares

There is more. We now consider solving minimization problems with **Lagrange Multipliers**. For proofs see GIT 1.14 and the references mentioned there. The minimization we solved in Section 5 (5) was an example of an **unconstrained minimization** in which we found the smallest possible value of a function. But suppose there is a side condition, called a **constraint**, that must hold for all solutions. The Figure 6.1, taken from GIT, show the general idea for single condition. If the constraint condition is expressed the in form:

$$g(x) = 0 \tag{1}$$

then the minimum of the constrained problem

$$\min_{x \in \mathbb{R}^m} f(x) \quad \text{with} \quad g(x) = 0 \tag{2}$$

occurs at a stationary point of the *unconstrained function* function

$$u(x, \mu) = f(x) - \mu g(x) \tag{3}$$

where we must consider variations of x and μ; of course μ is called a Lagrange multiplier. If there are n constraint conditions in the from $g_k(x) = 0,\ k = 1, 2, \cdots n$, each would be associated with its own Lagrange multiplier:

$$u(x, \mu_1, \mu_2, \cdots \mu_n) = f(x) - \sum_{k=1}^{n} \mu_k g_k(x)\,. \tag{4}$$

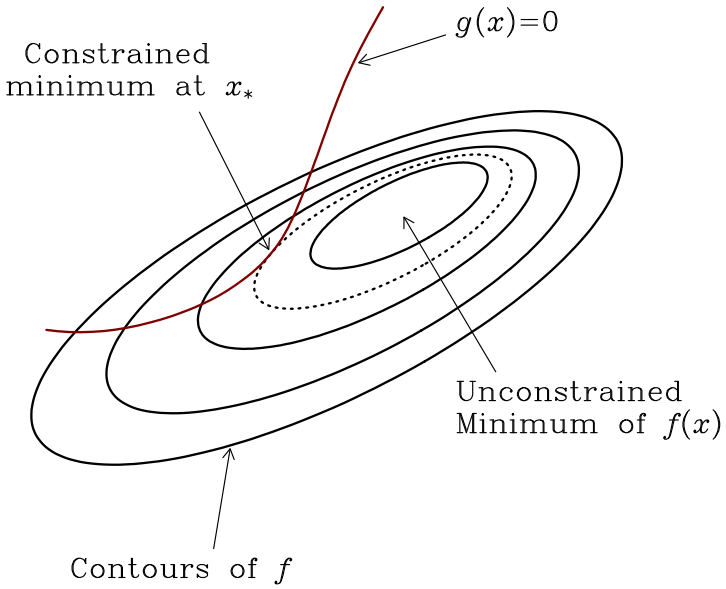


Figure 6.1: An optimization problem in 2 unknowns with 1 constraint.

As an example consider again the overdetermined LS problem solved by 5(13). We wish to find the minimum of the function $f(r) = \| r \|^2$, with $r \in \mathbb{R}^m$. As an unconstrained problem the answer is obviously zero. But we have the following m conditions on r:

$$0 = y - Ax + r \tag{5}$$

where $y \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$ are known, while the vector $x \in \mathbb{R}^n$ is also unknown and free to vary. So, writing (5) out in components and giving each row its own Lagrange multiplier, (4) becomes for this problem

$$u(r, x, \mu_l) = \sum_{j=1}^{m} r_j^2 - \sum_{j=1}^{m} \mu_j (y_j - \sum_{k=1}^{n} a_{jk} x_k + r_j) . \tag{6}$$

Differentiating over r_i, x_i and μ_i, the stationary points of u occur when

$$\frac{\partial u}{\partial r_i} = 0 = 2r_i - \mu_i \tag{7}$$

$$\frac{\partial u}{\partial x_i} = 0 = - \sum_{j=1}^{m} a_{ji} \mu_j \tag{8}$$

$$\frac{\partial u}{\partial \mu_i} = 0 = y_i - \sum_{k=1}^{n} a_{ik} x_k + r_i . \tag{9}$$

Equation (7) says the vector of Lagrange multipliers $\mu = 2r$; then using this fact and translating (8), (9) into matrix notation:

$$A^T \mu = 2A^T r = 0 \tag{10}$$

$$Ax - r = y . \tag{11}$$

If we multiply (11) from the left with A^T and use (10) we get the normal equations 5(12) again. But let us do something else: we combine (10) and (11) into a single linear system in which the unknown consists of both x and r:

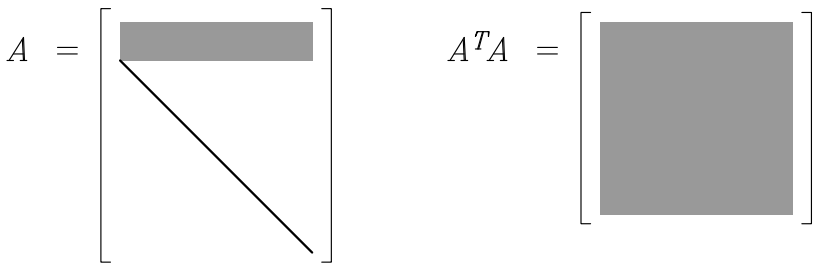


Figure 6.2: An example of loss of sparseness in forming the normal equations.

$$\begin{bmatrix} -I_m & A \\ A^T & O_n \end{bmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix} \tag{12}$$

where $I_m \in \mathbb{R}^{m \times m}$ is the unit matrix, where $O_n \in \mathbb{R}^{n \times n}$ is square matrix of all zeros. This system has the same content as the normal equations, but solves for the residual and the coefficients at the same time. If A is sparse, (12) can be a better way to solve the LS problem than by the normal equations or by QR, particularly as QR does not have a good adaptation to sparse systems. The situation is illustrated for a common form of overdetermined problem in Figure 6.2.

We turn next to the so-called **underdetermined least-squares** problem. While the overdetermined LS problem occurs with monotonous regularity in statistical parameter estimation problems, the underdetermined LS problem looks quite a lot like an *inverse problem*. Instead of trying to approximate the known y by a vector in the column space of A, we can match it exactly: we have

$$y = Ax\,. \tag{13}$$

where $A \in \mathbb{R}^{m \times n}$, but now $m < n$ and A is of full rank. This is a finite-dimensional version of the linear forward problem, in which the number of measurements, y, is less than the number of parameters in the model x. So instead of looking for the smallest error in (13), which is now zero, we ask instead for the *smallest model, x*. We are performing a simplified regularization, in which size, here represented by the Euclidean length, stands for simplicity. This problem is solved just as the last one, with a collection of m Lagrange multipliers to supply the constraints given by (13), but with $\|x\|^2$ being minimized instead of $\|r\|^2$. I'll run through the process quickly. The unconstrained function is

$$u(x, \mu_k) = \sum_{j=1}^{n} x_j^2 - \sum_{i=1}^{m} \mu_i \left(\sum_{k=1}^{n} a_{ik} x_k - y_i \right) \tag{14}$$

$$\frac{\partial u}{\partial x_j} = 2x_j - \sum_{i=1}^{m} a_{ij} \mu_i \tag{15}$$

$$\frac{\partial u}{\partial \mu_i} = -\sum_{k=1}^{n} a_{ik} + y_i\,. \tag{16}$$

Setting these derivatives to zero leads to a pair of linear systems which can be written in matrix notation

$$x = \tfrac{1}{2} A^T \mu \tag{17}$$

$$Ax = y \tag{18}$$

Equation (17) contains a key piece of information: the norm minimizer lies in the range space of A or, in other words, x is a linear combination of the column vectors of A. If we substitute the first of these into the second

we have

$$½AA^T\mu = y\ . \tag{19}$$

which we imagine solving somehow, then substituting for μ in the first member of (17)

$$x = A^T(AA^T)^{-1}y\ . \tag{20}$$

These are the normal equations for the underdetermined (smallest norm) problem. Explicitly following equation (20) is rarely a good way to compute the solution, however. First, if matrix A is sparse we loose that property forming AA^T: then it is better to combine (17) and (18) into a large sparse system combining μ and x in a longer unknown vector as we did in (12). Second, we can use QR to find a numerically stable result.

Like the normal equations, (19) too suffers from poor conditioning numerically. And as before QR comes to the rescue, but in a cute way. Recall that the classic QR factorization works only if $m \geq n$, here that is violated. So we write instead that

$$A^T = QR \quad \text{or} \quad A = R^TQ^T\ . \tag{21}$$

Then (13) can be written

$$y = R^TQ^Tx = R^T\hat{x} \tag{22}$$

where $\hat{x} \in \mathbb{R}^m$ is just Q^Tx. Then, since Q is an orthogonal matrix

$$x = Q\hat{x} \tag{23}$$

and it follows that x and $\hat{x}$ have the same norm, ie. Euclidean length. Recall that R is upper triangular; so (22) is

$$y = [R_1^T \ \ O]\begin{pmatrix}\hat{x}_1\\ \hat{x}_2\end{pmatrix} \tag{24}$$

where $R_1 \in \mathbb{R}^{n\times n}$, and $\hat{x}_1 \in \mathbb{R}^n$. If we multiply out the partitioned matrix we see that

$$y = R_1^T\hat{x}_1 + O\,\hat{x}_2 = R_1^T\hat{x}_1\ . \tag{25}$$

Because the second term vanishes, (25) shows that we can choose $\hat{x}_2$ (the bottom part of $\hat{x}$) in any way we like and it will not affect the match to the data: only $\hat{x}_1$ influences that. So we match the data exactly by solving the system

$$R_1^T\hat{x}_1 = y\ . \tag{26}$$

Now observe that

$$\|x\|^2 = \|\hat{x}\|^2 = \|\hat{x}_1\|^2 + \|\hat{x}_2\|^2\ . \tag{27}$$

So to match the data we solve (26), then to get the smallest norm we simply put $\hat{x}_2 = 0$. Thus $\hat{x}$ has been found that minimizes the norm, and the corresponding x is recovered from (23).

The underdetermined LS problem is artificial in the sense that (13) the condition that the model fit the data *exactly* is unrealistic: if there is noise in the data y, we must not demand an exact fit. It is more realistic to say that we would be satisfied with a reasonably close fit, as measured by the Euclidean norm; so replace (13) with

$$|| Ax - y || \leq \gamma \tag{28}$$

where we get choose γ from a statistical criterion that depends on the noise in y. Problems with a single inequality constraint turn out to be very similar to those with equality constraints. One of two scenarios can apply: (a) the unconstrained problem satisfies (28); (b) equality holds in (28), in which case a Lagrange multiplier can be used for the minimization. For the moment, let us concentrate on (b), which is the usual state of affairs. We need a single Lagrange multiplier to apply (28). To complicate things slightly more, instead of minimizing the norm of x, we will minimize

$$f(x) = || Px ||^2 \tag{29}$$

where $P \in \mathbb{R}^{p \times n}$ is a matrix that suppresses undesirable properties, for example, it might difference x to minimize slopes instead of magnitudes. Now we have the unconstrained function

$$u(x, \mu) = || P\, x ||^2 - \mu(\gamma^2 - || Ax - y ||^2) \tag{30}$$

where I have squared the condition factor because it will simplify things later. A trivial rearrangement gives:

$$u(x, \mu) = || P\, x ||^2 + \mu || Ax - y ||^2 - \mu\gamma^2 . \tag{31}$$

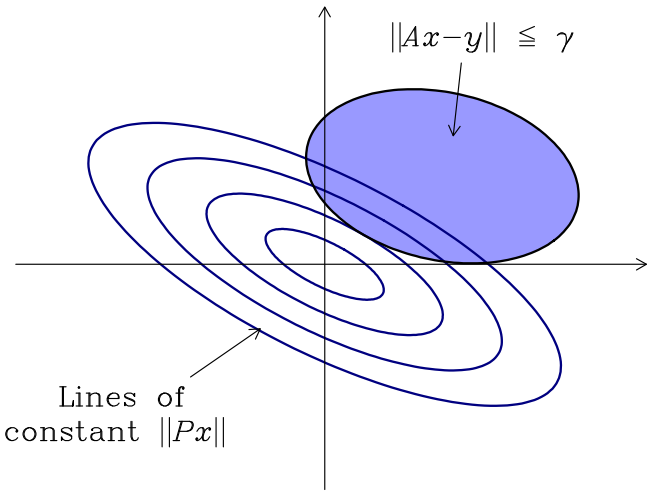


Figure 6.3: Minimization of $|| Px ||$ subject to $|| Ax - y || \leq \gamma$ for $x \in \mathbb{R}^2$.

It can be shown (see GIT, Chapter 3) that $\mu > 0$. Then for a fixed value of μ, the function u can be interpreted as finding a compromise between two undesirable properties, large $P\,x$, and large data misfit. If we minimize over x with a small μ we give less emphasis to misfit and find models that keep $P\,x$ very small; and conversely, large μ causes minimization of u to yield x with small misfit. This is an example of a **trade-off** between two incompatible quantities: it is shown in GIT that decreasing the misfit always increases the penalty norm, and vice versa.

We could solve this problem by differentiating in the usual tedious way. Instead we will be a bit more clever. As usual, differentiating by μ just gives the constraint (28). The derivatives on x don't see the γ term in (31) so we can drop that term when we consider the stationary points of u with respect to variations in x:

$$\hat{u}(x) = \| P\,x \|^2 + \mu \| Ax - y \|^2 \tag{32}$$

$$= \| Px - 0 \|^2 + \| \mu^{½} Ax - \mu^{½} y \|^2 . \tag{33}$$

Both of the terms are norms acting on a vector; we can make the sum into a single squared norm of a longer vector, the reverse of what we did on equation 5(24):

$$\hat{u}(x) = \left\| \begin{bmatrix} P \\ \mu^{½} A \end{bmatrix} x - \begin{pmatrix} 0 \\ \mu^{½} y \end{pmatrix} \right\|^2 \tag{34}$$

$$= \| Cx - d \|^2 . \tag{35}$$

The matrix $C \in \mathbb{R}^{(p+m)\times n}$ must be tall, that is $p + m > n$, or the original problem has a trivial solution (Why?), (35) is just an ordinary *overdetermined least squares problem*; indeed the matrix C is the one illustrated in Figure 6.2. So for any given value of μ, we can find the corresponding x through our standard LS solution. But this doesn't take care of (28). The only way to satisfy this misfit criterion is by solving a series of versions of (35) for different guesses of μ in an iterative way, because unlike all the other systems we have met so far, this equation is nonlinear. We will discuss the details later (see GIT, Chap 3).

What about scenario (a)? We need to verify that the unconstrained problem, the minimizer of $\| Px \|^2$ satisfies (28). When P is nonsingular, that is easy, because then the unique solution is $x = 0$, and that can be checked trivially in (28). If P is singular the solution to the unconstrained problem is not unique, and we could set up the minimization of $\| Ax - y \|$ over the null space of P. But in fact we will discover in solving (31) that (28) is satisfied for all $\mu > 0$ as part of the search in μ, so solving (28) with an equality is all we need ever to do.

7. Other Matrix Factorizations

The QR factorization is one of a number of matrix factorizations that appear in the numerical analysis of linear algebra. The rule that numerical analysts repeat with great regularity is that to solve the linear system

$$Ax = y \tag{1}$$

never, never, never calculate the inverse A^{-1} and multiply this into the vector y. The reasons are that it is numerically inaccurate, and inefficient. The recommended way is via one of several factorizations. To solve (1) in MATLAB you should always type:

```
x = A \ y;
```

Recall that when $A \in \mathbb{R}^{m\times n}$ and the problem is overdetermined ($m > n$), this gets you the least-squares solution via QR. When $m = n$, the system is solved, with QR, but by **Gaussian elimination** which can also be written as a matrix factorization called **LU** decomposition:

$$A = L\,U\,. \tag{2}$$

Here $A, L, U \in \mathbb{R}^{n\times n}$ and L is lower triangular, while U is upper triangular, called U because for some unknown reason the word 'upper' is always used here instead of 'right' (which is always the name used in QR). Formally the solution to (1), once you have the LU factors is to solve by back substitute the two systems

$$Lz = y, \quad \text{and} \quad Ux = z\,. \tag{3}$$

You don't need to know this of course just to use back-slash. Unlike QR factorization, LU decomposition of a sparse matrix A results in two sparse factors L and U, which is a very important property for large systems of equations.

A special case arises when A is symmetric, and positive definite. Then A can be factored with the **Cholesky factorization**

$$A = L\,L^T \tag{4}$$

where L is lower triangular. Notice that this factorization is almost like a square root of A, and is handy in a number of situation when a matrix square root could be useful. Cholesky factorization is one of the fastest and must numerically stable schemes in numerical linear algebra.

Next let me briefly remind you about the elementary theory of eigenvalue systems for square matrices. You will recall that a square symmetric matrix always has **eigenvalues**, which are the real numbers λ satisfying

$$Au = \lambda u, \quad \text{and } u \neq 0\,. \tag{5}$$

When $A \in \mathbb{R}^{n\times n}$ is not symmetric λ need not be real, and in some cases there are no solutions to (5); the symmetric case covers almost all those of practical interest. When A is symmetric, there are at most n distinct values of λ, call them λ_k, and n corresponding **eigenvectors** u_k.

Conventionally, these are normalized so that $\| u_k \| = 1$, in the 2-norm. A most important property of the eigenvectors is that they are mutually orthogonal:

$$u_j^T u_k = 0, \quad \text{when} \quad j \neq k \,. \tag{6}$$

When there are fewer than n eigenvalues, the system is said to be **degenerate** and can be treated as if there are repeated values of λ; and then the eigenvectors of the degenerate eigenvalues are not uniquely defined. But they can always be chosen to be orthogonal so that (6) can be forced to be true, and always is in computer programs. The simplest illustration of all this is the unit matrix: every vector is an eigenvector of I with eigenvalue 1; so the eigensystem is n-fold degenerate, that is, there are n eigenvalues, all the same, all equal to one.

The eigenvalue problem can be written as a matrix factorization, as we shall now see. Form the square matrix U from columns of the orthogonal eigenvectors:

$$U = [u_1, u_2, \cdots u_n] \tag{7}$$

The matrix U is an orthogonal matrix, because its columns are mutually orthogonal unit vectors. Then

$$AU = [Au_1, Au_2, \cdots Au_n] = [\lambda_1 u_1, \lambda_2 u_2, \cdots \lambda_n u_n] = U\Lambda \tag{8}$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots \lambda_n)$. Now multiply on the right with U^T and we have the **spectral factorization** of A:

$$A = U\Lambda U^T \tag{9}$$

This can be written another way that is most instructive:

$$A = \lambda_1 u_1 u_1^T + \lambda_2 u_2 u_2^T + \cdots \lambda_n u_n u_n^T \tag{10}$$

$$= \lambda_1 P_1 + \lambda_2 P_2 + \cdots \lambda_n P_n \tag{11}$$

Here the outer product matrices $u_k u_k^T$ are projection matrices each of which maps a vector into the subspace comprised of the corresponding eigenvector. Equation (11) is decomposing the action of A into components in an orthogonal coordinate system, where each component receives a particular magnification by the corresponding λ_k. Think about what this means when there is degeneracy. It should be observed that numerical techniques for discovering the eigenvalues and eigenvectors of symmetric matrices are based on performing the factorization (9), not on evaluating some huge determinant, which would take an eternity.

There is a spectral factorization for nonsquare matrices as well, called **singular value decomposition** usually called **SVD.** Suppose $A \in \mathbb{R}^{m \times n}$ with $m > n$, then A can be factorized into the product of three matrices, two orthogonal and one diagonal:

$$A = U \Sigma V^T \tag{12}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal and $\Sigma \in \mathbb{R}^{m\times n}$ with

$$\Sigma = \mathrm{diag}(s_1, s_2, \cdots s_n) \tag{13}$$

The real numbers $s_k \geq 0$ are called the **singular values** of A. One can solve LS problems with SVD. There are number of people who claim SVD is the answer to almost all LS problems because of its great numerical stability and because of its use in censoring the poorly resolved features in simple minimization problems. I believe these advantages are usually overstated; also the procedure is numerically very expensive, and not readily adapted to sparse systems, and therefore not suitable for large systems.