Excursions in Computing Science: Week 7c. Coordinates, Angles and Reality

T. H. Merrett*
McGill University, Montreal, Canada

August 2, 2024

I. Prefatory Notes

A. Reality

Gonna jump down, spin around, pick a bale of cotton.
Gonna jump down, spin around, pick a bale a day.
Norman Luboff, Harry Belafonte and William Attaway

1. Vectors are real.
   - Independent of coordinate axes, so
   - transform in a certain way when we change the axes.

Example transformations:

$$\begin{aligned} \text{rotate} \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\ \text{reflect } x \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} -1 & \\ & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \end{aligned}$$

Note two assumptions underlying this Week: all the coordinate systems considered have common origin and common units.

So what are *not* vectors?

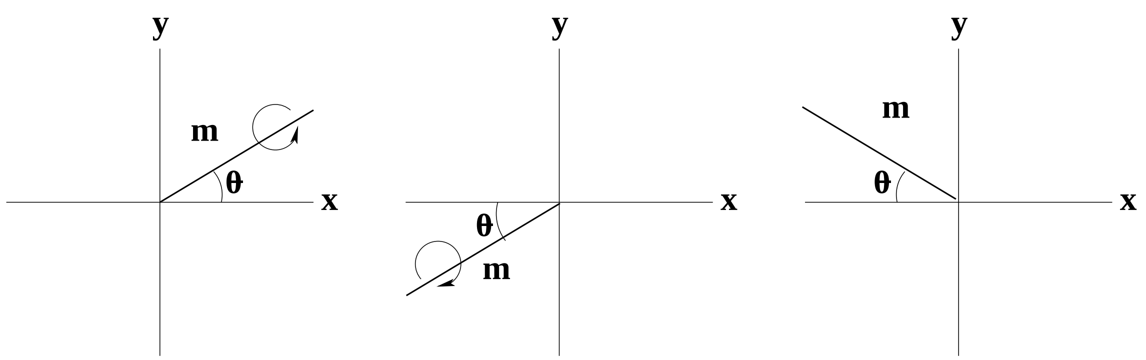
A twirl is not:

it has magnitude m and direction θ,

so $x = \cos\theta$ and $y = \sin\theta$

but it does not reflect the way a vector does.

 The author gratefully acknowledges support from the taxpayers of Québec and of Canada who have paid his salary and research grants while this work was developed at McGill University, and from his students and their funding agencies.

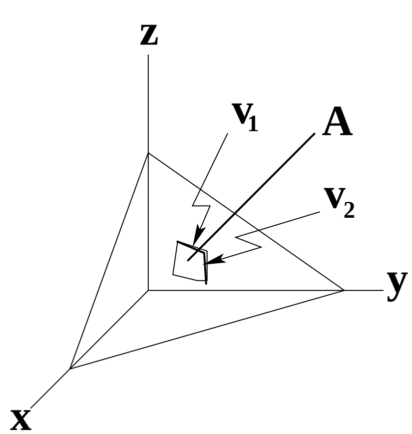


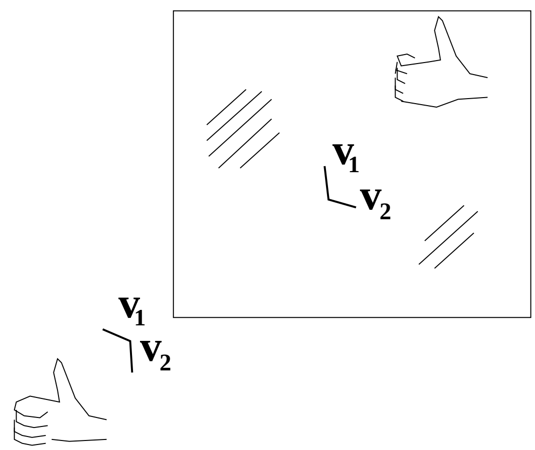
Twirl (m, θ) **Reflected x→ −x** **This is not the reflection**

We get $m' = -m$, i.e., $x' = -x$ and $y' = -y$

instead of $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} -1 & \\ & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$

In 3D, an *area* is like a twirl: it can have an *orientation* to distinguish above from below.





We saw that a right-handed twirl becomes a left-handed twirl in the mirror.

Similarly the direction of turn needed to rotate v_1 into v_2 is reversed in the mirror. This direction can be taken to determine the orientation of the parallelopiped area defined by v_1 and v_2.

In some sense, $v_1 v_2 = -v_2 v_1$: the "product" is anticommutative. We'll follow up this essential insight shortly (Note 6).

2. Some pairs are not vectors: their components are not coordinates.

$$\begin{pmatrix} \text{apples}' \\ \text{oranges}' \end{pmatrix} \stackrel{??}{=} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \text{apples} \\ \text{oranges} \end{pmatrix}$$

This is not a totally hokey example. Information retrieval (I.R.) often uses "vectors" to capture the content of documents.

	around	bale	cotton	day	down	jump	pick	spin	
doc1(	1	1	1	0	1	1	1	1	)
doc2(	1	1	0	1	1	1	1	1	)

I.R. even uses dot products (Week 2, Note 5) to detect similarity between documents:

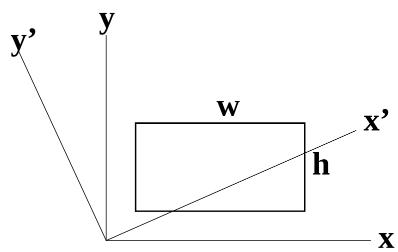
$$(\text{doc1} \ . \ \text{doc2})/(|\text{doc1}||\text{doc2}|) = 6/(\sqrt{7}\sqrt{7}).$$

But documents are not vectors: it is not meaningful to rotate or reflect the axes.

3. Even pairs of numbers from geometry, where rotating and reflecting *are* meaningful, are not always vectors. Let's try

$$\begin{pmatrix} \text{height} \\ \text{width} \end{pmatrix}$$

Here, no matter what the axes do, these numbers should not change.



What kind of thing remains invariant no matter what the axes do?

As with a vector, this thing, this pair of numbers, has a reality independent of the choice of coordinate axes. But the components of this one do not change if axes are rotated or reflected.

How about a matrix whose *eigenvalues* are w and h?

$$\begin{aligned} T\vec{v_1} &= w\vec{v_1} \\ T\vec{v_2} &= h\vec{v_2} \end{aligned}$$

For example, given the axes x and y shown,

$$\begin{aligned} T &= \begin{pmatrix} w & \\ & h \end{pmatrix} \\ v_1 &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ v_2 &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

Then, for axes x' and y', related to x and y by rotation R,

$$\vec{v_1'} = R\vec{v_1} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and

$$RTR^{-1}\vec{v_1'} = RTR^{-1}R\vec{v_1} = RT\vec{v_1} = Rw\vec{v_1} = wR\vec{v_1} = w\vec{v_1'}$$

This suggests that T transforms to the new axes as $T' = RTR^{-1}$.

$$\begin{aligned} \text{Hence } T'\vec{v_1'} &= w\vec{v_1'} \\ \text{Similarly } T'\vec{v_2'} &= h\vec{v_2'} \end{aligned}$$

This is called a *tensor* transformation. Height and width (almost) form a "tensor". This tensor is a diagonal matrix, $\begin{pmatrix} w & \\ & h \end{pmatrix}$, when the axes are aligned with the rectangle, as x and y are.

This tensor is not diagonal for all coordinate axes, but we can see that it is a symmetric matrix.

$$T' = RTR^{-1} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} w & \\ & h \end{pmatrix} \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$$

A symmetric matrix, T, equals its own transpose, $T = T^T$.

In general we may think of a tensor loosely as a matrix describing some real thing, as opposed to an operation or transformation.

$T' = RTR^{-1}$ is symmetric because the inverse of R is the transpose of R, $R^{-1} = R^T$, which is the case for rotations, reflections and other "orthogonal" transformations of coordinate axes.

4. Maybe twirl is a tensor too.

Try $S = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ and reflect in y by reversing the direction of x using the reFlection matrix F to give the tensor transformation FSF^{-1}

$$-\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} -1 & \\ & 1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} -1 & \\ & 1 \end{pmatrix} = \begin{pmatrix} a & -b \\ -c & d \end{pmatrix}$$

(Remember, Note 1 found out that the reflection just changes the sign of the twirl, i.e., of the tensor representing it.)

So $a = 0 = d$.

Any reflection will give a similar sign change, so let's see what reflecting in the line $x = y$ gives us: $F = \begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$

$$-\begin{pmatrix} & b \\ c & \end{pmatrix} = \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \begin{pmatrix} & b \\ c & \end{pmatrix} \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} = \begin{pmatrix} & c \\ b & \end{pmatrix}$$

and so $c = -b$.

Unfortunately, we've gone too far. We now have only one number, b, to describe a twirl, which we saw in Note 1 requires two numbers, m and θ.

So maybe two dimensions is too small to contain a twirl. This rather makes sense now that we think of it.

Let's see if we can decribe a twirl in three dimensions.

First note that $\begin{pmatrix} & -b \\ b & \end{pmatrix}$ is an *antisymmetric* matrix: it equals the negative of its transpose.

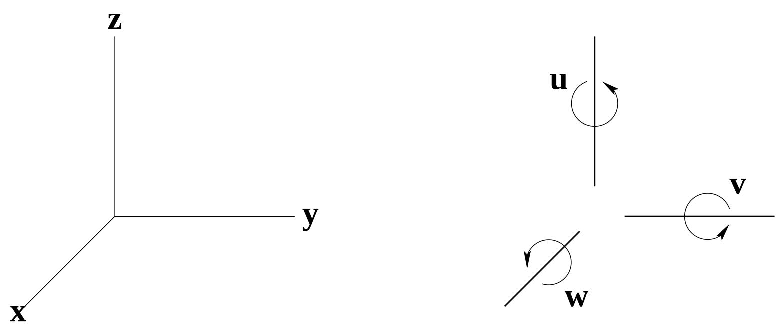
So we'll try an antisymmetric matrix in 3D. A 3×3 antisymmetric matrix has three components.

$$\begin{pmatrix} & u & v \\ -u & & w \\ -v & -w & \end{pmatrix}$$

Try reflecting in the yz plane: $x \leftrightarrow -x$

$$\begin{pmatrix} -1 & & \\ & 1 & \\ & & 1 \end{pmatrix} \begin{pmatrix} & u & v \\ -u & & w \\ -v & -w & \end{pmatrix} \begin{pmatrix} -1 & & \\ & 1 & \\ & & 1 \end{pmatrix} = \begin{pmatrix} & -u & -v \\ u & & w \\ v & -w & \end{pmatrix}$$

This almost just changes the sign of the matrix. Is it right?



Yes, if we interpret w as the x-component of the twirl, v as the y-component and u as the z-component. Check the diagram carefully!

Let's see what happens if we rotate in the xy plane.

$$\begin{pmatrix} c & s & \\ -s & c & \\ & & 1 \end{pmatrix} \begin{pmatrix} & u & v \\ -u & & w \\ -v & -w & \end{pmatrix} \begin{pmatrix} c & -s & \\ s & c & \\ & & 1 \end{pmatrix} = \begin{pmatrix} & u & sw+cv \\ -u & & cw-sv \\ -(sw+cv) & -(cw-sv) & \end{pmatrix}$$

This should be, and is, the same result we would get with $\begin{pmatrix} w \\ v \\ u \end{pmatrix}$ being just a vector, transformed in the usual vector way,

$$\begin{pmatrix} c & -s & \\ s & c & \\ & & 1 \end{pmatrix} \begin{pmatrix} w \\ v \\ u \end{pmatrix}$$

So a twirl, while transforming like a vector under rotation, is in general a tensor; for instance, it does not transform like a vector under reflection.

(Even though "twirl" is in one sense a rotation, we are here looking at it as a "real thing" so the matrix representing it is a tensor—as opposed to the quite different matrix that describes the *operator*, rotation.)

5. Twirl and area are "pseudovectors" or "axial vectors" in Willard Gibbs' vector analysis (which is widely used in spatial science). We now know that they are really tensors. It is just a coincidence that 3×3 antisymmetric tensors have 3 components, like a vector. This does not happen in two dimensions (1 component) or four dimensions (6 components).

Vector analysis generates pseudovectors by a "cross product" of two vectors: $A = v_1 \times v_2 = -v_2 \times v_1$, to use the area example from Note 1.

Vector analysis is unsatisfactory because

a) it is not a closed system: operating on vectors we get things that are not vectors (and, worse, they *look like* vectors);

b) it only works in three dimensions and does not generalize to more, or fewer, dimensions.

Can we make better abstractions for spatial entities, instead of vectors?

We need a formalism

- which is independent of coordinate axes;
- which captures the notion of area being the anticommutative combination of two vectors;
- which does not depend on the number of dimensions of the space.

B. Interval Algebra

6. Vectors and Areas and .. All Together

- Parts of space are lines, areas, volumes, ..
- We'll ignore absolute position and consider only direction and magnitude.
- We'll take the basis elements to be orthonormal and anticommutative.

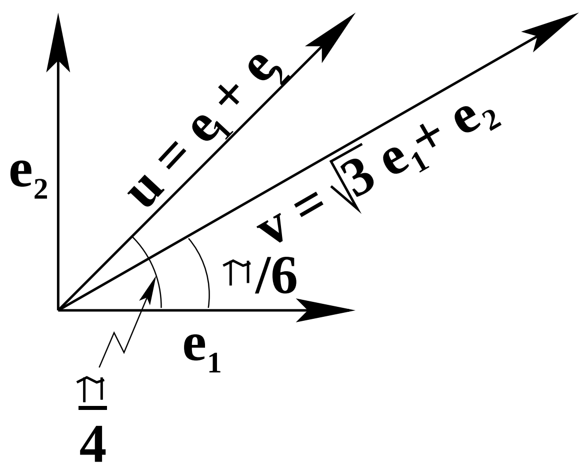
(We'll use the word "elements" instead of "vectors": some but not all elements can be thought of as vectors.)

1. The basis elements are e_1 and e_2, which are defined to have the following properties.

$$\begin{aligned} e_1e_1 &\stackrel{\text{def}}{=} 1 \\ e_2e_2 &\stackrel{\text{def}}{=} 1 \\ e_{12} &\stackrel{\text{def}}{=} e_1e_2 \stackrel{\text{def}}{=} -e_2e_1 \end{aligned}$$

2. An arbitrary element can be a linear combination of basis elements. Its product with itself is the square of its length or magnitude.

$$\begin{aligned} u &= e_1 + e_2 \\ uu &= (e_1 + e_2)(e_1 + e_2) = 1 + 1 = 2 \\ v &= \sqrt{3}e_1 + e_2 \\ vv &= (\sqrt{3}e_1 + e_2)(\sqrt{3}e_1 + e_2) = 3 + 1 = 2^2 \end{aligned}$$



3. The product of two different elements gives their magnitudes times the cosine and sine of the angle between them.

$$\begin{aligned} uv &= (e_1 + e_2)(\sqrt{3}e_1 + e_2) \\ &= \sqrt{3} + 1 + (1 - \sqrt{3})e_{12} \\ &= 2\sqrt{2}\Big(\frac{\sqrt{3}+1}{2\sqrt{2}} + \frac{1-\sqrt{3}}{2\sqrt{2}}e_{12}\Big) \\ &= 2\sqrt{2}(\cos(\pi/6 - \pi/4) + \sin(\pi/6 - \pi/4)e_{12}) \\ (ce_1 + se_2)(c'e_1 + s'e_2) &= (cc' + ss') + (cs' - c's)e_{12} \\ &= \cos(v - u) + \sin(v - u)e_{12} \end{aligned}$$

where $\cos(v-u)$ and $\sin(v-u)$ are respectively the cosine and sine of the angle from u to v: an *interval* from a to b is $b-a$ because adding s to it gives b.

7. Rotation

Let's have a magnitude operator ($|\, v \,|$ is an alternative notation),

$$\text{mag}(v) = |\, v \,| = \sqrt{vv} = \text{length of } v$$

and a normalizing operator (${}^n v$ is an alternative notation),

$$\text{norm}(v) = {}^n v = v/\text{mag}(v): \ \text{norm}(v)\text{norm}(v) = 1; \ v \, \text{norm}(v) = \text{mag}(v)$$

and $\text{norm}(v)\text{norm}(u)u = \ \text{norm}(v)\text{mag}(u)$, which rotates u into the direction of v.

$$\begin{aligned} \text{Try norm}(u) &= ce_1 + se_2 \\ v &= \text{mag}(v)(c'e_1 + s'e_2) = xe_1 + ye_2 \\ \text{norm}(u)\text{norm}(v) &= (cc' + ss') + (cs' - sc')e_{12} \\ &= C + Se_{12} \end{aligned}$$

where $C = \cos(v-u)$ and $S = \sin(v-u)$ as in Note 6. Compare this with 2-numbers, $C + iS$.

If we note that $e_{12}e_{12} = e_1e_2e_1e_2 = -e_1e_2e_2e_1 = -1$, we seem to find that e_{12} is the square root of -1. It's better to think of e_{12} as a $\pi/2$ rotation when postmultiplied (or a $-\pi/2$ rotation when premultiplied):

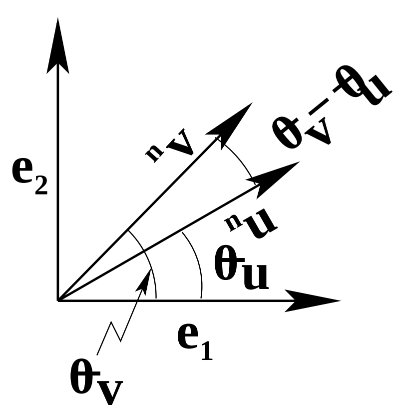
$$\begin{aligned} e_1e_{12} &= e_2 & \qquad e_{12}e_2 &= e_1 \\ e_2e_{12} &= -e_1 & \qquad e_{12}e_1 &= -e_2 \end{aligned}$$

(It is even better to think of e_{12} as a *plane*: see Note 11, below.)

So what is the meaning of $C + Se_{12}$?

$$\begin{aligned} u(C - Se_{12}) &= (xe_1 + ye_2)(C + Se_{12}) \\ &= (Cx - Sy)e_1 + (Sx + Cy)e_2 \\ &= (e_1 \ e_2)\begin{pmatrix} C & -S \\ S & C \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \end{aligned}$$

It's the rotation that rotates u onto v (the figure uses ${}^n v$ for $\text{norm}(v)$): $u\text{norm}(u)\text{norm}(v) = v = \text{norm}(v)\text{norm}(u)u$.

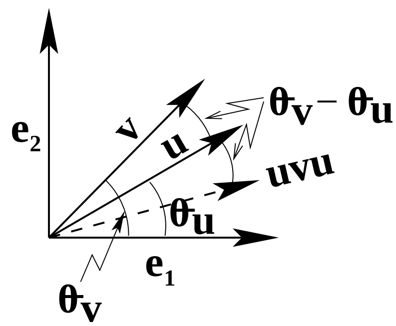


8. Reflection

If uuv and vuu rotate $u \to v$ what is uvu?

Let's try it with u and v normalized.

$$
\begin{aligned}
u &= c'e_1 + s'e_2 \\
v &= ce_1 + se_2 \\
uvu &= (c'e_1 + s'e_2)(ce_1 + se_2)(c'e_1 + s'e_2) \\
&= Ce_1 + Se_2 \text{ where} \\
C &= \cos(\theta_u - \theta_v + \theta_u) = \cos(\theta_u - (\theta_v - \theta_u)) \\
S &= \sin(\theta_u - \theta_v + \theta_u) = \sin(\theta_u - (\theta_v - \theta_u))
\end{aligned}
$$



uvu is the *reflection* of v in u.

(Another viewpoint: since $w(vu)$ rotates w by the angle between v and u, so $u(vu)$ is the reflection of v in u.)

Note that the *projection* of v in u is $(uvu + v)/2$, which can be written as a relationship among the reflection operator, F, the identity operator, I, and the projection operator, P: $P = (F + I)/2$.

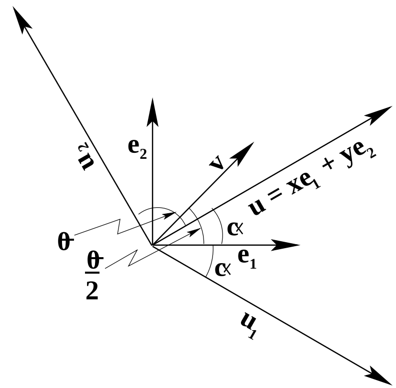
Note finally that a rotation is two reflections:

1. in e_1;
2. in "half-u", an element whose angle with e_1 is half the angle we wish to rotate through.

(We'll use the subscript J to indicate half-angles, since J sort of looks like 2 upside-down.)

$$
\begin{aligned}
v &= xe_1 + ye_2 & c = \cos\theta \quad c_J = \cos\theta/2 \\
u_J &= c_J e_1 + s_J e_2 & s = \sin\theta \quad s_J = \sin\theta/2 \\
u_J e_1 v e_1 u_J &= (c_J - s_J e_{12}) v (c_J + s_J e_{12}) \\
&= (e_1 \; e_2) \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}
\end{aligned}
$$

which is the rotation. (Recall that $c = c_J^2 - s_J^2$ and $s = 2c_J s_J$.) $\theta/2 + \theta/2 + \alpha - \alpha = \theta$:



9. 3D rotations

Outside of a 2-D plane we can't use $C + Se_{12}$ in 3-D:

$$e_3(C + Se_{12}) = Ce_3 + Se_{123}$$

(Note the extension of the rule for combining basis elements:

$$e_3e_{12} = e_3e_1e_2 = -e_1e_3e_2 = e_1e_2e_3 \stackrel{\text{def}}{=} e_{123})$$

So let's try two reflections:

$$\begin{aligned} \text{rotate } v &= xe_1 + ye_2 + ze_3 \\ \text{in plane } P &= re_{12} + pe_{23} + qe_{31} \end{aligned}$$

with P normalized: $p^2 + q^2 + r^2 = 1$.

$(c_J - s_JP)v(c_J + s_JP) =$

$$(e_1\ e_2\ e_3)\left(\begin{pmatrix} c & -sr & sq \\ sr & c & -sp \\ -sq & sp & c \end{pmatrix} + (1-c)\begin{pmatrix} p \\ q \\ r \end{pmatrix}(p, q, r)\right)\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Note that $pe_1 + qe_2 + re_3 \perp P = re_{12} + pe_{23} + qe_{31}$.

Note also that $\begin{pmatrix} p \\ q \\ r \end{pmatrix}$ is an eigenvector of the rotation matrix: what is the significance of that?

Now *two* rotations:

$$\begin{aligned} \text{by } (c, s) \text{ about } \quad & pe_1 + qe_2 + re3 \\ \text{then by } (c', s') \text{ about } \quad & p'e_1 + q'e_2 + r'e3 \\ & \Downarrow \\ \text{a rotation by } (c'', s'') \text{ about } \quad & p''e_1 + q''e_2 + r''e3 \end{aligned}$$

$$\begin{aligned} &(c_J + s_J(re_{12} + pe_{23} + qe_{31}))(c'_J + s'_J(r'e_{12} + p'e_{23} + q'e_{31})) \\ &= c''_J + s''_J(r''e_{12} + p''e_{23} + q''e_{31}) \end{aligned}$$

where

$$\begin{aligned} c''_J &= c_Jc'_J - s_Js'_J(rr' + pp' + qq') \\ s''_Jr'' &= s_Jc'_Jr + c_Js'_Jr' + s_Js'_J(qp' - pq') \\ s''_Jp'' &= s_Jc'_Jp + c_Js'_Jp' + s_Js'_J(rq' - qr') \\ s''_Jq'' &= s_Jc'_Jq + c_Js'_Jq' + s_Js'_J(pr' - rp') \end{aligned}$$

Note that in 3-D all the angles are half angles.

Note that 3-D rotations do not commute.

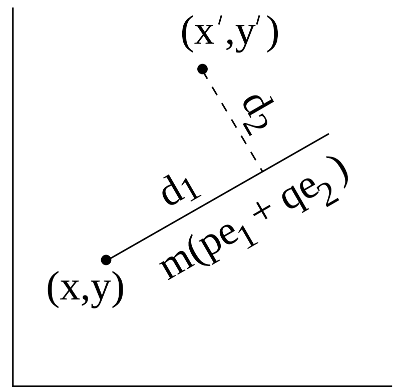
10. Intervals plus locations. The intervals described by the interval algebra have magnitude and orientation but no location.

Thus they cannot solve problems such as finding the distance from a point to a line.

We must work with both points and line intervals.

For a start, the line interval must be anchored to a point, say (x, y).

Then we can formulate the problem as "find the distance from a point (x', y') to the line that is the interval $m(pe_1 + qe_2)$ starting from point (x, y).



Here, m is the magnitude of the interval and p and q give its orientation (normalized so $p^2+q^2 = 1$). (x, y) and (x', y') are two points, which are beyond the scope of the interval algebra, and our task is to find the length of the dashed line, which is the distance from (x', y') to the interval starting at (x, y), and to ascertain that this vertical actually meets the original line within the interval.

We can find the dashed line as an interval, $p'e_1+q'e_2$, which we might as well normalize, $p'^2+q'^2 = 1$. This does not locate the interval, but we can do that by making it start at (x', y'). For orthogonality, either $pp' + qq' = 0$ or, directly, take the interval product with the containing plane

$$p'e_1 + q'e_2 = (pe_1 + qe_2)e12 = -qe_1 + pe_2$$

We consider as unknowns the two distances to the intersection of the solid and dashed lines, d_1 from (x, y), and d_2 from (x', y').

Then we can switch to vector notation and write

$$\begin{pmatrix} x \\ y \end{pmatrix} + d_1 \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix} - d_2 \begin{pmatrix} -q \\ p \end{pmatrix}$$

(Why is there a $-$ sign before d_2?)

This becomes

$$\begin{pmatrix} p & -q \\ q & p \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} x' - x \\ y' - y \end{pmatrix}$$

which is easily solved, especially since the determinant is 1.

If $d_1 \le m$, the dashed line does meet the solid line within the given interval.

11. Interval algebra in 3D. We can get a better feeling for interpreting the interval algebra from using it in 3D than from using it in 2D.

First, let's define the *order* of an interval, or part of an interval, as the number of subscripts on the basis elements. Intervals can be of homogenous order, such as the line interval $pe_1 + qe_2 + re_3$

(order 1) or the plane interval $pe_{23} + qe_{31} + re_{12}$ (order 2), or of mixed order, such as the product $pp' + qq' + rr' + (pq' - qp')e_{23} + (qr' - rq')e_{31} + (rp' - pr')e_{12}$.

In three dimensions, order-1 intervals are line intervals and order-2 intervals are plane intervals. The magnitude of a line interval is its length.

A plane interval can be considered equally as a plane area (the "area" is its magnitude) or as a twirl. In either case. it can also be thought of, in 3D, as the line interval orthogonal to it ($pe_{23}+qe_{31}+re_{12}$ or $pe_1 + qe_2 + re_3$), and in the twirl case, this line interval is the axis of rotation.

For those familiar with Gibbs' vector notation, this orthogonal to a plane interval is the "pseudovector" that is the "cross product" of the two vectors in the plane whose magnitudes times the sine of the angle between them is the magnitude of the cross product. It is much better to think of plane intervals than pseudovectors. For one thing, intervals generalize to any number of dimensions. Pseudovectors do not.

It is also useful to be able to extract components of any given order, so we define a **cmpt** operator with a parameter specifying the order of the component to be extracted.

$$\mathbf{cmpt}(0, (pe_1 + qe_2 + re_3)(p'e_1 + q'e_2 + r'e_3) = pp' + qq' + rr'$$

and similarly for $\mathbf{cmpt}(2, (pe_1 + qe_2 + re_3)(p'e_1 + q'e_2 + r'e_3))$, etc.

The Gibbs' "dot product" between the two vectors is given by $\mathbf{cmpt}(0,)$ and the "cross product" is given by $\mathbf{cmpt}(2,)$.

Now, the interval product between two homogeneous intervals of the same order gives the angle between them and the plane common to them.

$$uv = \mathrm{mag}(u)\mathrm{mag}(v)(c + s\,\mathrm{normPlane}(u, v))$$

For example,

$$\begin{aligned}(pe_1 + qe_2 + re_3)(p'e_1 + q'e_2 + r'e_3) &= pp' + qq' + rr' + (pq' - qp')e_{23} + (qr' - rq')e_{31} + (rp' - pr')e_{12} \\ &= -(pe_{23} + qe_{31} + re_{12})(p'e_{23} + q'e_{31} + r'e_{12})\end{aligned}$$

where

$$\begin{aligned}\mathrm{mag}(u) &= 1 = \mathrm{mag}(v) \\ c &= pp' + qq' + rr' \\ s &= \sqrt{(pq' - qp')^2 + (qr' - rq')^2 + (rp' - pr')^2}\end{aligned}$$

and $\mathrm{normPlane}(u, v)$ is the normalized plane consisting of the second-order component divided by s.

The interval product between a homogeneous interval and a containing interval gives the interval within the containing interval that is orthogonal to the first interval. Here is a slightly more general example.

$$\begin{aligned}(p'e_1 + q'e_2 + r'e_3)(pe_1 + qe_2 + re_3) &= (pp' + qq' + rr')e_{123} + (pq' - qp')e_3 + (qr' - rq')e_1 + (rp' - pr')e_2 \\ &= (p'e_{23} + q'e_{31} + r'e_{12})(pe_1 + qe_2 + re_3)\end{aligned}$$

where, if the line (order-1) intervals are contained, respectively, in the plane (order-2) intervals, $pp' + qq' + rr' = 0$ because the line intervals are orthogonal, respectively, to the line intervals orthogonal ("normals") to the plane intervals. (The word "normals" can be confused with the word "normal", describing an interval of magnitude 1, so we do not continue to use it.)

We can use these interpretations to find a new set of orthonormal axes (orthogonal to each other and normalized) given one desired axis. We work a specific example in which $f_1 = (e_1+e_2+e_3)/\sqrt{3}$.

First, find the plane interval orthogonal to f_1

$$(e_1 + e_2 + e_3)e_{123}/\sqrt{3} = (e_{23} + e_{31} + e_{12})/\sqrt{3}$$

Second, find any line interval in this plane: the condition is that $p'e_1 + q'e_2 + r'e_3$ is orthogonal to f_1 so $p' + q' + r' = 0$. This eliminates one of the three unknowns, and we might as well make a choice among the others which is as simple as possible. So suppose $r' = 0$ and $q' = -p'$:

$$f_2 = (e_1 - e_2)/\sqrt{2}$$

Third, find a second line interval in the orthogonal plane which is orthogonal to the first.

$$\begin{aligned} f_3 &= (e_1 - e_2)(e_{23} + e_{31} + e_{12})/\sqrt{3}/\sqrt{2} \\ &= (e_1 + e_2 - 2e_3)/\sqrt{6} \end{aligned}$$

Check that $f_j f_j = 1$ and $f_j f_k = -f_k f_j$ if $j \neq k$, the same properties that the e_j have..

Finally, observe that the matrix transforming from the e_j to the f_k is just given by the coefficients.

$$\begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{2} & \\ 1/\sqrt{6} & 1/\sqrt{6} & -2/\sqrt{6} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

Check that the inverse of this matrix is its transpose, and convince yourself that the transformation of *coordinates*, if the *space* were to be rotated the same way, relative to the original axes, e_j, is this transpose.

12. Summary

(These notes show the trees. Try to see the forest!)

- Vectors are real things, independent of coordinates.
- So where they are written in terms of coordinates, these coordinates must transform correctly under rotation, reflection, projection and inversion: $X\vec{v}$.
- *Some* real things are not vectors, but tensors, and so tensor elements must also transform correctly: XTX^{-1}.
- Clifford or geometric or angle or interval algebra:
  - parts of space: lines, areas, volumes, ..;
  - ignore position, consider only magnitude, direction;
  - basic elements are orthonormal and commutative.
- 2-D rotation from u to v is uuv or vuu.
- Reflection of v in u is uvu.
- 3-D rotation by (c, s) about $re_{12} + pe_{23} + qe_{31}$ ..
- Two 3-D rotations need half angles and are not commutative.
- Intervals have no locations, only magnitudes and orientations, so the interval algebra must be supplemented by points if, say, distances are to be found.
- Interval products have a number of useful interpretations, including angles between lines and planes, and orthogonals to lines and planes.

NB In 2-D: $1, e_1, e_2, e_3, e_{12}$. In 3-D: $1, e_1, e_2, e_3, e_{23}, e_{31}, e_{12}, e_{123}$.

13. Appendix: Summary of vector and matrix operations

$+$

$$\begin{aligned}\vec{u}+\vec{v} &= \begin{pmatrix} u_1+v_1 \\ u_2+v_2 \end{pmatrix} \\ A+B &= \begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{pmatrix}\end{aligned}$$

$\bullet$

$$\begin{aligned}\vec{u}.\vec{v} &= (u_1\ u_2)\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\ &= u_1v_1+u_2v_2 \\ &= \mid \vec{u} \mid\mid \vec{v} \mid \cos(\angle(\vec{u},\vec{v})) \\ A\vec{u} &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \\ &= \begin{pmatrix} a_{11}u_1+a_{12}u_2 \\ a_{21}u_1+a_{22}u_2 \end{pmatrix} \\ \vec{u}A &= (u_1\ u_2)\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \\ &= (u_1a_{11}+u_2a_{21}\ \ u_1a_{12}+u_2a_{22}) \\ AB &= \begin{pmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{pmatrix}\end{aligned}$$

$\otimes$

$$A\otimes B = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix}$$

Clifford algebra

$$\begin{aligned}uv &= (u_1e_1+u_2e_2)(v_1e_1+v_2e_2) \\ &= u_1v_1+u_2v_2+(u_1v_2-u_2v_1)e_{12} \\ &= \vec{u}.\vec{v}+\mid \vec{u}\times\vec{v} \mid e_{12} \\ &= \text{mag}(u)\text{mag}(v)(\cos(\angle(\vec{u},\vec{v}))+\sin(\angle(\vec{u},\vec{v}))e_{12})\end{aligned}$$

(The third line does not use the Clifford algebra mag() operator because it is not Clifford algebra. It is a digression for those familiar with Gibbs' vector algebra.)

Compare $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}(v_1,v_2) = \begin{pmatrix} u_1v_1 & u_1v_2 \\ u_2v_1 & u_2v_2 \end{pmatrix}$

Finally, compare these with 2-numbers (Week 4: we use 2-number notation for the magnitude instead of the Clifford algebra mag() operator):

$$\begin{aligned}u+v &= u_1+v_1+i(u_2+v_2) \\ uv &= (u_1+iu_2)(v_1+iv_2) \\ &= u_1v_1-u_2v_2+i(u_1v_2+u_2v_1) \\ &= \mid u \mid e^{i\angle u} \mid v \mid e^{i\angle v} \\ &= \mid u \mid\mid v \mid e^{i(\angle u+\angle v)}\end{aligned}$$

II. The Excursions

You've seen lots of ideas. Now *do* something with them!

1. **Dot product.**
   a) The dot product (see Week 2 Note 5) of two normalized vectors in any number of dimensions equals the cosine of the angle between the vectors. Show this: i) use $(Xu)^T Xv = u^T v$ to discover that the dot product is invariant under any axis transformation, X, whose transpose is its inverse; and ii) use this invariance to reduce any two d-dimensional vectors, $\vec{u}$ and $\vec{v}$, to the two dimensions of their common plane.
   b) What is the angle between doc1 and doc2 in Note 2?
   c) How does the dot product of a vector, v, with itself relate to the interval algebra product vv in Note 6?
   d) How would you define the dot product of the interval algebra basis elements e_1 and e_2 so that for any vectors $u = u_1 e_1 + u_2 e_2$ and $v = v_1 e_+ v_2 e_2$, $u.v = u_1 v_1 + u_2 v_2$? Relate this to the definition of the dot product (Week 2 Note 5) in terms of some particular coordinate system. Why is the product $a_x b_x + a_y b_y$ of any coordinates a_x, a_y, b_x and b_y invariant, i.e., has the same value no matter what axes are used to specify the coordinates?
   e) Show that $\vec{u}.\vec{v}/ \mid v \mid$ is the component of $\vec{u}$ along the direction of $\vec{v}$ for any two vectors $\vec{u}$ and $\vec{v}$.

2. Calculate the reflections in the yz plane of twirls pointing along each of the x, y and z axes, and explain why what you get is right.

3. Confirm that w, u and v in the 3D twirl tensor must refer to the x, y and z components, respectively.

4. Is there a way to use 2-numbers to represent 3D twirl as a 2×2 tensor?

5. Show that postmultiplyimg by e_{12} is the same as premultiplying by e_{21} in Note 7. What does this imply for expressing the rotation from u to v as a premultiplication?

6. What is the matrix for the reflection of $v = xe_1 + ye_2$ in $u = ce_1 + se_2$ (c and s are cosine and sine, respectively, so u is normalized)?

7. Why is $u(vu)$ the reflection of v in u? Explain in terms of the rotation, (uv). (Take u and v to be normalized.)

8. A ball moving along trajectory b bounces off a wall w. What is its new trajectory?

9. Explain why the projection of v on u is $(uvu + v)/2$. For $u = c'e_1 + s'e_2$ and $v = ce_1 + se_2$, give the matrices F (reflection) and P (projection). What is the significance of $P - I$, where I is the identity matrix?

10. Show that 3D rotation by angle (c, s) about $re_{12} + pe_{23} + qe_{31}$ is the matrix given in Note 9. Show that $(p, q, r)^T$ is an eigenvector (Note 1 of Week 8), find the corresponding eigenvalue, and explain what these mean.

11. Check the derivation of the expression for double rotation in 3D. How would we find p'', q'' and r''?

12. Compare rotating by $\pi/2$ about (1,0,0) then $\pi/2$ about (0,1,0) with rotating $\pi/2$ about (0,1,0) then $\pi/2$ about (1,0,0). Use both interval algebra and your hands and some physical object such as a book.

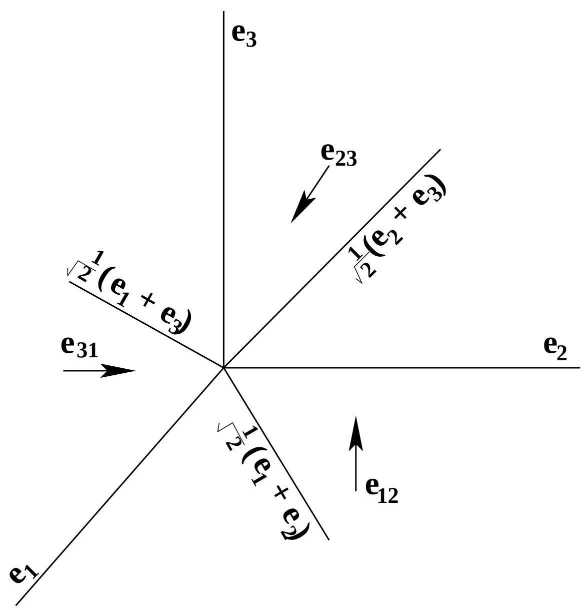
13. Using rotations (and other operations) in the interval algebra and a starting edge, e_1, find the other two edges of an equilateral triangle. How would this help you draw it with a graphics program?
Once you've found the second edge, there are at least three ways of finding the third: figure them all out and compare.

14. Rotate the equilateral triangle of the previous Excursion just enough to map it onto itself and show that the edges you found there do indeed map onto each other.

15. Why can the Interval Algebra not be used to find the intersection of two lines?

16. **Tetrahedron.** Using rotations (and other operations) in the interval algebra and the equilateral triangle of the previous Excursion, calculate the three edges needed to build it into an equilateral tetrahedron. How would you find the angles between the planes in the tetrahedron?

17. What is the 3-by-3 matrix that gives a 1/3 rotation (i.e., by $2\pi/3$) about the axis (1,1,1)? Check that this make sense: multiply it by itself once, then once more.



18. a)
(Warmup and check.) What is the plane formed by the edges $\mathbf{e}_1$ and $(\mathbf{e}_2 + \mathbf{e}_3)/\sqrt{2}$? What is the angle between these two edges? What angle does the plane make with $\mathbf{e}_{12}$? (Keep all edges and planes normalized! Be careful about signs, and check what they mean!)
b) Answer the questions from (a) for the edges $(\mathbf{e}_1 + \mathbf{e}_3)/\sqrt{2}$ and $(\mathbf{e}_2 + \mathbf{e}_3)/\sqrt{2}$.

c) Examine and test the MATLAB function

```
% function [cos12,sin12,face12] = product(edge1,edge2)
% THM 070410    in file: product.m
% edge1: normalized 3-vector, e.g. [p1,q1,r1]
% edge2: normalized 3-vector, e.g. [p2,q2,r2]
% cos12 = p1p2+q1q2+r1r2
% sin12 = +sqrt(1-cos^2)
% face12: normalized 3-vector,
%  [(q1r2-r1q2)/sin12,(r1p2-p1r2)/sin12,(p1q2-q1p2)/sin12]
% (Works for planes as input, but use -cos12, -sin12)
```

```
function [cos12,sin12,face12] = product(edge1,edge2)
p1 = edge1(1); q1 = edge1(2); r1 = edge1(3);
p2 = edge2(1); q2 = edge2(2); r2 = edge2(3);
cos12 = p1*p2+q1*q2+r1*r2;
sin12 = sqrt(1-cos12^2):  % when might this be 0?
if abs(sin12)<10^-8 face12 = [0,0,0]; else
face12 = [(q1*r2-r1*q2)/sin12,(r1*p2-p1*r2)/sin12,(p1*q2-q1*p2)/sin12];
end
```

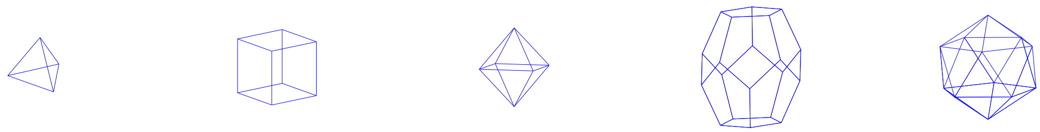
Why must we change the sign if `edge1` and `edge2` represent faces rather than edges on input? (Hint. Multiplying by $\mathbf{e}_{12}$ in 2D gives a quarter-rotation. Does multiplying by $\mathbf{e}_{123}$ in 3D also do this? What does a "quarter rotation" mean in this case for an edge? For a face? What is $\mathbf{e}_{123}\mathbf{e}_{123}$?)
d) (Warmup and check.) Rotate the edges $\mathbf{e}_1$ and $(\mathbf{e}_2 + \mathbf{e}_3)/\sqrt{2}$ through the angle you found in (a) so as to put them both in $\mathbf{e}_{12}$: this should give $\mathbf{e}_1$ itself and $\mathbf{e}_2$, respectively.
e) Rotate the edges from (b) so as to put them both in $\mathbf{e}_{12}$. Check that they have the same angle with each other that they did before rotating.
f) Find two additional normalized edges that share with each of the new edges from (e) the same angle you found in (b) that they have with each other. (Note that the solution is direct if the input edges are in $\mathbf{e}_{12}$ but would require iteration if the $\mathbf{e}_3$ components of the edges are nonzero: try it!)
g) Write a MATLAB function, `e12equiAngle()`, for (f), i.e., which given two edges in $\mathbf{e}_{12}$ finds an edge sharing with those two edges the angle that is between the input edges.
Write a MATLAB function, `equiAngle()`, which given *any* two edges finds an edge sharing with those two edges the angle that is between the input edges: find the plane of the given edges, rotate it into the $\mathbf{e}_{12}$ plane, use `e12equiAngle()` to find the new edge, and rotate this back again. (The next excursion gives a possible `rotate3D()` function interface.)
h) Rotate the edge from (f) that has the negative $\mathbf{e}_3$ component inversely to the rotation in (e). What is the resulting combination of this edge and the two original edges in (b)?

19. Inspect and run the following MATLAB function.

```
% function [pentcoords,pentedges,pentface] = pentagon(startcoords,startedge,pentface)
% THM 070409    in file: pentagon.m
% Makes pentagon of unit edges, given 3D coords for 1 vertex, 1 edge, 1 plane
% startcoords 3-vector, e.g. [0,0,0]
% startedge 3-vector, e.g. [1,0,0]
% pentface 3-vector, e.g.[0,0,1]      The plane in which the pentagon is made
% pentcoords 5*3 array, e.g. [0,0,0;1,0,0;..]
% pentedges 5*3 array, e.g. [1,0,0;..]
% uses rotate3D
function [pentcoords,pentedges,pentface] = pentagon(startcoords,startedge,pentface)
angle = 2*pi/5;
edgesIN = startedge';
planesIN = pentface';
pentedges = edgesIN;
pentcoords = startcoords';
[edgesOUT,planesOUT] = rotate3D(pentface,angle,edgesIN,planesIN);
for k = 1:4
   pentedges = [pentedges,edgesOUT];
   coords = pentcoords(k,:) + pentedges(k,:)
```

```
    pentcoords = [pentcoords,coords];
    [edgesOUT,planesOUT] = rotate3D(pentface,angle,edgesOUT,planesIN);
  end
```

Write the function `rotate3D(plane,angle,edgesIN,planesIN)`, which rotates arbitrary sets of `edgesIN` and `planesIN` about `angle` in `plane`.
Write a program which calls `pentagon()` and uses `quiver3` to draw the resulting pentagon.

20. 

Above are the five "Platonic solids": the tetrahedron (4 faces), the cube (hexahedron, 6 faces), octahedron (8 faces), dodecahedron (12 faces) and icosahedron (20 faces). Use the techniques of the previous excursions to build them in MATLAB.
(The cube and octahedron do not need interval algebra machinery and their edges can be written down straight from pairs of coordinates. They make a good place to start. The tetrahedron can also be written down directly from coordinates, or it can be made from an equilateral triangle and an additional vertex out of the plane and equidistant from each vertex of the triangle; but it is good exercise to use interval algebra for this, following the **Tetrahedron** Excursion, above, or the notes on Clifford Algebra available from the course home page.)
By finding a way to draw the octahedron inside the cube and the icosahedron inside the dodecahedron, show that these are two pairs of "duals"—the faces of one of each pair correspond to the vertices of the other, and vice-versa. What is the dual of the tetrahedron?

21. Use the pdf notes "Clifford Algebra" for this week to find the coordinates of the centre of a tetrahedron (the point equidistant from each vertex) and to show that the angle between any two edges connecting the centre with two vertices is about $109^o27'$.

22. Look up William Kingdon Clifford, 1845–1879, and describe his role in creating the interval algebra. (It is really called the Clifford algebra, or sometimes the geometric algebra.)

23. How many colours are needed to colour the vertices of each of the Platonic solids, if no two vertices of the same colour may be joined by an edge? How many colours for the faces, if no two faces separated by an edge as a boundary may have the same colour? What about colouring vertices of polygons in 2D?

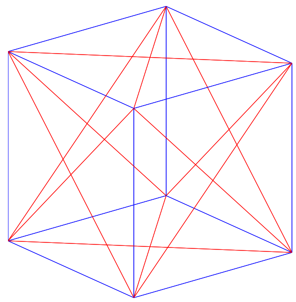
24. Confirm that the Platonic solids satisfy

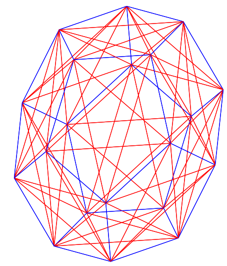
$$2 + E = F + V$$

where E is the number of edges, F is the number of faces and V is the number of vertices. Does this hold for any other figure?

25. How many spheres can be packed around a sphere of the same radius? (Hint: start with 2D and show that six circles pack a centre circle. What angle does each circle subtend at the centre? Approximately what proportion of the spherical surface area, $4\pi r$, is inside one of the packing spheres centred at distance r? Must the centres of the packing spheres form the vertices of one of the Platonic solids?)

26.





The red additions to the cube and the dodecahedron above are the paths of length 2. That is, since the cube has a blue edge (0,0,0)–(1,0,0) and a blue edge (1,0,0)–(1,1,0), then (0,0,0)–(1,1,0) will be a red edge.
Here are all the coordinate pairs for the cube, in two different orders: the set on the left is sorted by columns 4, 5 and 6; the set on the right is sorted by columns 1, 2 and 3.

0	0	1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	0	0
0	1	1	0	0	1	0	0	1	0	1	1
1	0	1	0	0	1	0	0	1	1	0	1
0	0	0	0	1	0	0	1	0	0	0	0
0	1	1	0	1	0	0	1	0	0	1	1
1	1	0	0	1	0	0	1	0	1	1	0
0	0	1	0	1	1	0	1	1	0	0	1
0	1	0	0	1	1	0	1	1	0	1	0
1	1	1	0	1	1	0	1	1	1	1	1
0	0	0	1	0	0	1	0	0	0	0	0
1	0	1	1	0	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	0	1	0	0	1
1	0	0	1	0	1	1	0	1	1	0	0
1	1	1	1	0	1	1	0	1	1	1	1
0	1	0	1	1	0	1	1	0	0	1	0
1	0	0	1	1	0	1	1	0	1	0	0
1	1	1	1	1	0	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	1	1	0	1
1	1	0	1	1	1	1	1	1	1	1	0

a) Confirm that these coordinate pairs link up so as to give the red edges shown with the cube.
b) Examine the following MATLAB code which will make the links you checked in (a). It implements a simplified *natural composition* operator of the *relational algebra*. It is built in terms of three other relational algebra operators, *natural join, projection* and a family of operators that treat relations as sets of rows and produce set difference ($-$), union (`u`), intersection (`n`) and symmetric difference ($+$). (Note that this last operator is here applied to the set of *columns* of the relations being put together.)
Look up [Mer99, Database programming], implement these operators, and show that

relationCompos() applied to the coordinate pairs for the cube produces the red figures shown.
c) Run your relationCompos() on the coordinate pairs you got for the dodecahedron in an earlier excursion.

```
% function joinOut = relationCompos(joinIndices,joinIn1,joinIn2)
% THM 070420 in file relationCompos.m
% joinIndices 2*m array giving indices to be joined
% joinIn1 n1*m1 array
% joinIn2 n2*m2 array
% joinOut n*(m1-m+m2) array
% joinOut rows will be unduplicated if joinIn1 and joinIn2 rows are
% Uses relationSetOp(), relationJoin(), relationProject()
function joinOut = relationCompos(joinIndices,joinIn1,joinIn2)
sizIn1 = size(joinIn1);
sizIn2 = size(joinIn2);
sizInd = size(joinIndices(1,:));
%all = zeros(sizInd);                % indices for compareRows(): all columns
for k = 1:sizIn1(2) - sizInd(2) + sizIn2(2) all(k) = k; end
projIndices = relationSetOp('-',all',joinIndices(1,:)')
joinOut = relationProject(projIndices',relationJoin(joinIndices,joinIn1,joinIn2));
```

27. Combine the methods of Notes 10 and 11 to find the distance in three dimensions between a given point and a line made up of an interval and another given point as start point. (You will need to solve for three distances, and combine two of them to give the desired distance.)

28. Note 11 gives the expression $c + s\ \mathrm{normPlane}(u, v)$.
a) Show that $c^2 + s^2 = 1$ so that it is plausible to interpret c and s as cos() and sin(), respectively. (Try it in two dimensions first.)
b) Use the axis-rotating method of Note 11 to change axes so that a new f_3 is orthogonal to the plane containing u and v, and thereby establish that c and s really are cos() and sin() in the two-dimensional f_{12} plane.

29. Show that $pp' + qq' + rr' = 0$ also results from the condition that the reflection of $p'e_1 + q'e_2 + r'e_3$ in the plane orthogonal to $pe_1 + qe_2 + re_3$ equals $p'e_1 + q'e_2 + r'e_3$ itself.

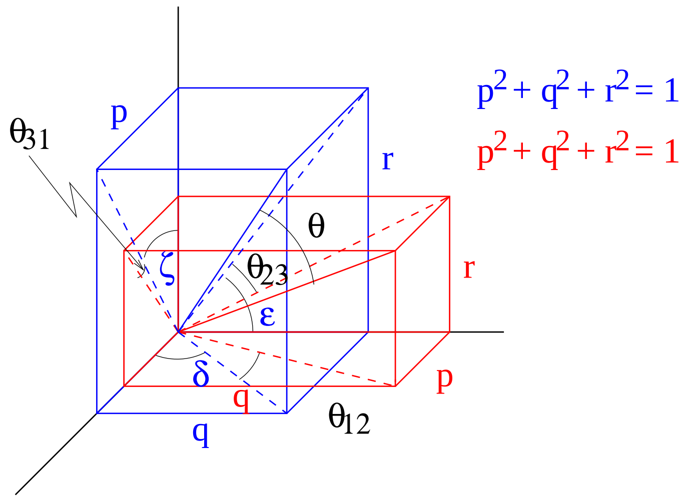
30. **Direction cosines.** The normalized p, q and r we have been using in Notes 9–11 for three dimensions are also known as direction cosines

$$p = \cos\alpha, \quad q = \cos\beta, \quad r = \cos\gamma$$

a) For line intervals, what are the angles α, β and γ?
b) What are the direction cosines in two dimensions (what must sin() be replaced by)?
c) When two line intervals, angle θ apart, are given by direction cosines, show that the sines of the projections of θ on the e_{12}, e_{23} and e_{31} planes are, respectively,

$$\sin\theta_{12} = \frac{pq' - qp'}{\sqrt{p^2+q^2}\sqrt{p'^2+q'^2}}, \quad \sin\theta_{23} = \frac{qr' - rq'}{\sqrt{q^2+r^2}\sqrt{q'^2+r'^2}}, \quad \sin\theta_{31} = \frac{rp' - pr'}{\sqrt{r^2+p^2}\sqrt{r'^2+p'^2}}$$

(In the figure, red is used for the primed direction cosines. Use the differences between the angles δ, ϵ and ζ shown and their red counterparts.)



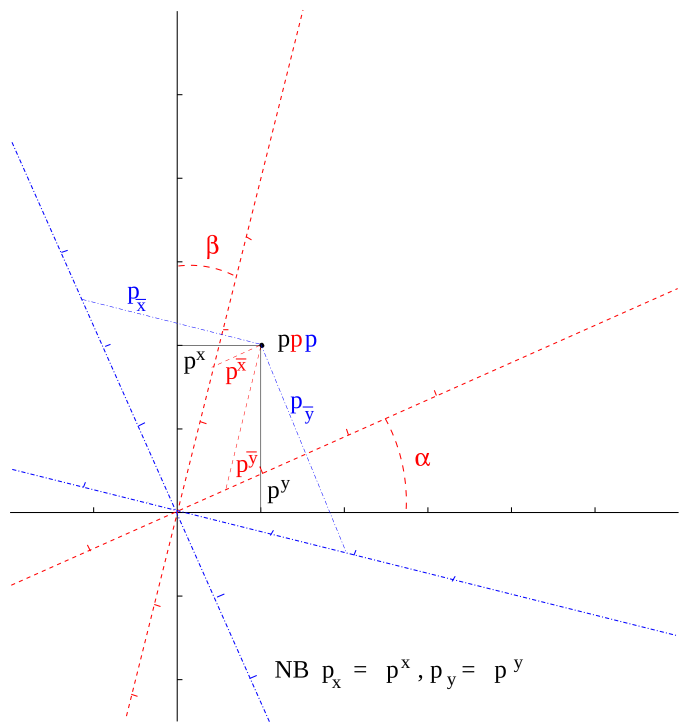
d) What are the cosines of these projections of θ?
e) Note that the redundancy of $p^2 + q^2 + r^2 = 1$ hides the signs. Why is it useful to have all three direction cosines? (All two in 2D?)

31. The Gibbs "cross product" of two vectors in 3D is defined as

$$(a, b, c) \times (a', b'c') = ((bc' - cb', ca' - ac', ab' - ba')$$

Show that this is not a vector but a tensor. Write the tensor.
Show that it is not a line interval but a plane interval. Write the plane interval.

32. **Nonorthogonal axes and tensor notation.** We've seen that the two "tensors" in Notes 3 and 4 are independent of rotations of the coordinate system. Tensor notation is intended to cope with *any* linear transformation of the axes. After the following discussion, show that the twirl of Note 4 survives *nonorthogonal* axis transformations but the height- and width-eigenvalues of Note 3 do not.
Here is a non-orthogonal axis transformation, from the black axes (solid lines) to the red (dashed lines).



a) To transform the vectors

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

to the red axes shown, persuade yourself that we would use the matrix

$$S = \frac{1}{\sqrt{c_+}} \begin{pmatrix} c_\alpha & s_\beta \\ s_\alpha & c_\beta \end{pmatrix}$$

where c_z is $\cos(z)$, s_z is $\sin(z)$ for z either α or β, and c_+ is $\cos(\alpha+\beta) = \cos(\alpha)\ \cos(\beta) - \sin(\alpha)\ \sin(\beta)$. (This latter is the determinant of the part of S written above as a matrix. We need not divide that matrix by $\sqrt{c_+}$ but to do so normalizes the matrix in a way that allows the example to illustrate a minor but significant point (see (j)).)
The first point to make is that to transform the *coordinates* describing the point $p = (p^x, p^y)$, so that the *same* point, p, is identified by new (red) coordinates, $p = (p^{\bar{x}}, p^{\bar{y}})$, we use the *inverse* of S,

$$S^{-1} = \frac{1}{\sqrt{c_+}} \begin{pmatrix} c_\beta & -s_\beta \\ -s_\alpha & c_\alpha \end{pmatrix}$$

b) Persuade yourself that this statement is true. (Think about rotations as an example, and compare finding a rotated point, p, with finding new coordinates for p under rotated *axes*.)
So

$$\begin{pmatrix} p^{\bar{x}} \\ p^{\bar{y}} \end{pmatrix} = \frac{1}{\sqrt{c_+}} \begin{pmatrix} c_\beta & -s_\beta \\ -s_\alpha & c_\alpha \end{pmatrix} \begin{pmatrix} p^x \\ p^y \end{pmatrix}$$

c) The new red coordinates are *non-orthogonal*: the red axes are not at right angles to each other. What is different about a non-orthogonal coordinate system is that invariants such as the length of a vector (say, the distance from p to the origin) or the angle between two vectors will apparently change under the transformation: calculate $\sqrt{p^{\bar{x}} \times p^{\bar{x}} + p^{\bar{y}} \times p^{\bar{y}}}$ and compare it to $\sqrt{p^x \times p^x + p^y \times p^y}$. This does not happen with orthogonal transformations such as rotations: such invariants are left safely fixed by the rotation. (Show that the two above square roots are the same if $\beta = -\alpha$ but not necessarily for other β. If your algebraic results do not convince you, try it with the two angles drawn in the above figure: $c_\alpha = 12/13$ and $c_\beta = 24/25$. Show that the squares of the lengths are about 2.75 versus 5. Or just look at the drawing.)

So we need to think of something else. This is the first contribution of tensor theory. Because we used S^{-1} to transform p^j to $p^{\bar{k}}$ this transformation is called *contravariant*. A corresponding transformation using S^T is called *covariant*, and that is what we need. Tensor notation writes contravariant elements with superscript indices. (That is why I repeated the p^x and the p^y above to square them instead of writing p^{x2} and p^{y2}: it is best in tensor notation not to use superscript operators to denote powers.)

And covariant elements are written with subscript indices. You'll find this in the blue components, $p_{\bar{x}}$ and $p_{\bar{y}}$, in the diagram.

Only in orthogonal coordinate systems are the contravariant and the covariant components the same. Thus, in black, $p_x = p^x$ and $p_y = p^y$.

In the diagram, the blue (dot-dash lines) shows the transformation.

$$\begin{pmatrix} p_{\bar{x}} \\ p_{\bar{y}} \end{pmatrix} = \frac{1}{\sqrt{c_+}} \begin{pmatrix} c_\alpha & s_\alpha \\ s_\beta & c_\beta \end{pmatrix} \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

(d) Show that the covariant transformation of (1,0) is orthogonal to the contravariant transformation of (0,1) and vice-versa in this example.

Now we consider how to describe the invariant length of p (that is, the distance of point p from the origin). We use *both* contravariant and covariant coordinate systems. Writing the transformations as matrices and the coordinates as vectors

$$(p_{\bar{x}}, p_{\bar{y}}) \begin{pmatrix} p^{\bar{x}} \\ p^{\bar{y}} \end{pmatrix} = (p_{\bar{x}}, p_{\bar{y}}) S S^{-1} \begin{pmatrix} p^{\bar{x}} \\ p^{\bar{y}} \end{pmatrix} = (p_x, p_y) \begin{pmatrix} p^x \\ p^y \end{pmatrix} = (p^x, p^y) \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

Here I used row vector $\times$ column vector to express the sum of products (in this case, the sum of squares), and I've used the less usual way of writing the transformation $S^T \times$(column vector) as (row vector)$\times S$.

Let's look at this in terms of the indices. I'll write them out first in the conventional matrix way with all indices as subscripts.

$$\begin{aligned}
\sum_j p_j p_j &= \sum_j (\sum_k p_k S_{kj})(\sum_{k'} S^{-1}_{jk'} p_{k'}) \\
&= \sum_k p_k \sum_{k'} (\sum_j S_{kj} S^{-1}_{jk'}) p_{k'} \\
&= \sum_k p_k (\sum_{k'} I_{kk'} p_{k'}) \\
&= \sum_k p_k p_k
\end{aligned}$$

From the first to the second line we rearranged the order of the sums, which you should convince yourself we can always do. Then we got SS^{-1}, and this is the identity I. The identity gets rid of the sum of k' by just setting k' to k.

Because we can always rearrange summation order and because multiplication of the individual elements commutes, even though multiplication of the matrices does not, tensors introduce a second notational simplification: the *Einstein summation convention* says drop the $\sum$ signs and *just repeat indices to sum*.

Combining this with the use of superscript indices for contravariant tensors and subscript indices for covariant, we can write this argument in tensor notation. Transposes don't appear at all. But I've written a place-holding dot just to indicate which was the left (row) and which the right (column) matrix index, in order to maintain a connection with matrix multiplication.

$$\begin{aligned}p_j p^j &= p_k S^k_{.j}(S^{-1})^j_{.k'}p^{k'}\\ &= p_k I^k_{.k'}p^{k'}\\ &= p_k p^k\end{aligned}$$

e) We saw that the second-order tensors in Notes 3 and 4 transform using both the transformation matrix and its inverse, e.g., $T \to RTR^{-1}$. Write this in tensor notation and argue that tensors that transform as T does should be written with one contravariant and one covariant index. That is why I wrote the transformation matrices, say S, above as S^k_j.
The above discussion requires that we have both contravariant and covariant components in order to compute invariants such as the inner product (length of one vector or angle between two different normalized vactors). But there is a way to find length, say, if we have only one of these sets, such as the contravariant components. We start with the length in the orthogonal system.

$$\begin{aligned}p^j p^j &= S^j_{.k}p^k S^j_{.k'}p^{k'}\\ &= p^k(S^T)^{.j}_k S^j_{.k'}p^{k'}\\ &= p^k(S^TS)_{kk'}p^{k'}\\ &= g_{kk'}p^k p^{k'}\end{aligned}$$

where $g_{kk'} \stackrel{\text{def}}{=} (S^TS)_{kk'}$ is the covariant double tensor called the *fundamental metric tensor*. g enables us to find invariants even though limiting ourselves to contravariant coordinates.
f) Calculate $g_{kk'}$ for the working example of this Excursion.
g) Show that $g_{kk'}$ is symmetric, i.e., $g_{kk'} = g_{k'k}$.
h) What is the *contravariant* fundamental metric tensor, $g^{kk'}$?
i) Show that $g_{kk'}$ applied to any tensor with a contravariant index k' (or k) *lowers* that index, making it covariant. Hint: use $p_{\bar{j}}p^{\bar{j}} = p_j p^j$.
At last, the "minor but significant point" I promised to illustrate back at the beginning of this Excursion when I introduced $1/\sqrt{c_+}$ as a normalizing factor in the example transformations.
j) Show that without that factor, S would transform (1,0) and (0,1) into vectors of the same length, 1. Then convince yourself that the scaling factor $1/\sqrt{c_+} = 1.13$ increases the distance between the unit marks along the transformed axes in the diagram.
k) The vector product $b_j x^j$ is not the only invariant. A constant, c, is always invariant, of course. And so is the 2nd-order product $g_{jk}x^j x^k$ above, as is, for *any* a_{jk}, $a_{jk}x^j x^k$.
These can be combined in a generalization of the quadratic equation for scalars $ax^2 + bx + c$ to the *general quadric*

$$a_{jk}x^j x^k + b_j x^j + c$$

What are the interpretations that can be made of the general quadric, in the sense that the quadratic can be interpreted as a parabola or, in special cases, a straight line? (Classify all the possibilities in 2D. Look into 3D: what are planes? What are lines? Show that any matrix is the sum of a symmetric and an antisymmetric matrix: what contribution to $a_{jk}x^j x^k$ is made by the antisymmetric part of a?)
l) Look up [McC57]: this Excursion prepares you for Parts I and II, where you can learn about classical geometry as an application of tensors. This is an older book but application-directed once you are over the initial hurdles—and you should now be prepared for these.
m) Without the normalizing factor $1/\sqrt{c_+}$ in the transformation, show that $g_{12} = g_{21} = s_+$ where s_+ is $\sin(\alpha+\beta) = \cos(\pi/2-(\alpha+\beta))$, the cosine of the angle between the nonorthogonal axes.

33. **Tensor Calculator I. Matrix representation.** One- and two-index tensors can be represented readily as vectors and matrices and the Einstein convention that repeated indices are summed over translates into regular matrix multiplication. Indices are either subscript (covariant) or superscript (contravariant: see previous Excursion) and raising or lowering them means (matrix) multiplication by the metric tensor or its inverse: since the metric tensor has two indices, either both down or both up, it can be included in the matrix representation employed in this Excursion. We build a MATLAB program to raise and lower indices and to multiply the matrices and vectors representing tensors.
We need a convention for the names of the matrices which distinguishes the positions of the indices, since, for instance the tensor $g_{\alpha,\beta}$ is different from $g^{\alpha,\beta}$—indeed, if it is the metric tensor, as its name implies in this example, one is the inverse of the other.
So we combine the name of the tensor with the position(s) of its index(es): $g_{\alpha,\beta}$, for example, is called `g_dd` because of the two "down" indices, and $g^{\alpha,\beta}$ is called `g_uu`.
We work with examples from [Har03, pp.423, 428]. For instance,

$$\texttt{g\_dd} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}$$

a) Show that

$$\texttt{g\_uu} = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix}$$

b) Because we must multiply a tensor by the appropriate metric tensor to raise or lower its indices,

$$t^{\alpha}_{\beta} = g^{\alpha\gamma} t_{\gamma\beta} \quad \text{or} \quad \texttt{t\_ud} = \texttt{g\_uu}\,\texttt{t\_dd}$$

and

$$t^{\beta}_{\alpha} = t_{\alpha\gamma} g^{\gamma\beta} \quad \text{or} \quad \texttt{t\_du} = \texttt{t\_dd}\,\texttt{g\_uu}$$

show that

$$\texttt{g\_ud} = \texttt{g\_uu}\,\texttt{g\_dd} = I$$

and

$$\texttt{g\_du} = \texttt{g\_dd}\,\texttt{g\_dd} = I$$

where I is the identity matrix.
c) Given, in addition to `g_dd` above, the single-index tensors

$$\begin{aligned} \texttt{a\_d} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \texttt{b\_d} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \texttt{c\_u} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \texttt{d\_u} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

show that

$$\begin{aligned} \texttt{a\_u} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \texttt{b\_u} &= \begin{pmatrix} 1 \\ -2 \end{pmatrix} \\ \texttt{c\_d} &= \begin{pmatrix} 2 \\ 1 \end{pmatrix} \\ \texttt{d\_d} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{aligned}$$

d) Given, also in addition to g_dd above, the double-index tensor

$$\texttt{t\_dd} = \begin{pmatrix} 3 & 1 \\ -1 & 0 \end{pmatrix}$$

show that

$$\begin{aligned} \texttt{t\_dd} &= \begin{pmatrix} -1 & 0 \\ 5 & 1 \end{pmatrix} \\ \texttt{t\_ud} &= \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} \\ \texttt{t\_du} &= \begin{pmatrix} 0 & -1 \\ 1 & 3 \end{pmatrix} \end{aligned}$$

We must build a table to associate each tensor name with its corresponding matrix (or vector). We can use a cell array:

```
  table = {'g_dd',[2,1;1,0];'a_d',[1;0];'b_d',[0;1]; 'c_u',[1;0];'d_u',[0;1];...
          't_dd',[3,1;-1,0]};
```

and write code which will extend this table each time we calculate a new tensor.

e) Start with a lookup function

```
function [name,array,d2u,u2d] = lookup(strg,table)
```

which uses `table` passively to find in the first column of `table` the `strg` and return in `name` the closest entry and in `array` the corresponding matrix. It should furthermore return in `d2u` the positions of the indices which must be raised to convert the found `name` to the sought `strg`, and in `u2d` the positions of the indices which must be lowered. Thus, with the above `table`

```
[name,array,d2u,u2d] = lookup3('t_dd',table)
```

should return the exact match

```
name = t_dd
array =
    3 1
   −1 0
d2u = []
u2d = []
```

while

```
[name,array,d2u,u2d] = lookup3('t_ud',table)
```

should return the nearest match (exact on “t” and closest on the sequence of “u”s and “d”s.

```
name = t_dd
array =
   3 1
   −1 0
d2u = [1]
u2d = []
```

and

```
[name,array,d2u,u2d] = lookup3('t_uu',table)
```

will return the same except

```
d2u = [1 2]
```

and finally

```
[name,array,d2u,u2d] = lookup3('x_dd',table)
```

reports that no tensor starting “x” is in `table`

```
name = ''
array = Inf
d2u = []
```

```
u2d = []
```

f) Now build the tensor calculator itself

```
function [answer,newTable] = tensorCalc(query,table)
```

which applies `lookup(query,table)` to give the suitably modified matrix as `answer` and also enters the new name and matrix into `newTable`. (You can go one further and let `query` be a sequence of `strg`s for `lookup()`, and, if the resulting matrices have suitably matching sizes, add at the end of `answer` the matrix product of the whole lot.) Thus,

```
[answer,newTable] = tensorCalc3('t_uu','c_d',table)
answer = [2x2 double] [2x1 double] [2x1 double]
newTable =
'g_dd' [2x2 double]
'a_d' [2x1 double]
'b_d' [2x1 double]
'c_u' [2x1 double]
'd_u' [2x1 double]
't_dd' [2x2 double]
'g_uu' [2x2 double]
't_uu' [2x2 double]
'c_d' [2x1 double]
answer:
ans =
   0 -1
   1 3
ans =
   2
   1
ans =
   -1
   5
```

Tensor notation is more general than matrices, so a matrix representation will not capture the full capabilities of tensors. For example, the above does not support dot product of two vectors, since vectors are all represented as columns (this could be patched, of course). It does not support contraction since that is expressed by repeated indices, which our representation is not capable of. And tensors of three or more indices will of course be awkward to express as conventional matrix multiplication.

34. **Tensor Calculator II. Relational representation.** A suitable generalization of matrices which can represent general tensors are relations, a computer data structure proposed by E. F. Codd in 1970 to rationalize databases on secondary storage but applicable much more generally and to matrices in particular. We must switch from MATLAB to the language Aldat [Mer07].

    A *relation* is a set of "n-tuples". It is a set in the mathematical sense, that it has no duplicate elements and that the order of the elements does not matter. For example, $\{a, b, b\}$ is not a set; $\{a, b, c\}$ is the same set as $\{a, c, b\}$ or as any permutation of these three letters. The order-independence of sets is a powerful abstraction which reduces the 3! (3-factorial) possible permutations of this example to a single entity.

    An *n-tuple* is a collection of n items for which order *does* matter, either absolutely, or relative to some pre-ordered list of names called *attributes*.

    Examples of relations representing vectors and matrices are

$$a_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad g_{uu} = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix} \qquad t_{dd} = \begin{pmatrix} -1 & 0 \\ 5 & 1 \end{pmatrix}$$

```
a_d(i  v)
    1  1
```

```
g_uu(i  j   v)
     1  2   1
     2  1   1
     2  2  −2
```

```
t_dd(i  j   v)
     1  1   3
     1  2   1
     2  1  −1
```

The indices are stored explicitly, as attributes named `i, j`, etc., and the value of each element has its own attribute named `v`. The relations each have a name, before the parentheses naming the attributes. Relation `a_d` is a "binary" relation: its elements being 2-tuples (or duples or pairs). Relations `g_uu` and `t_dd` are "ternary" relations, with 3-tuples (or triples) as elements, The data follows in columns below the names in each heading line. (This particular way of showing relations is called the "tabular representation".)
Here is a 3-index tensor, the "alternating tensor". Its elements are 4-tuples or quadruples.

```
alt_uuu(i  j  k   v)
        1  2  3   1
        1  3  2  −1
        2  3  1   1
        2  1  3  −1
        3  1  2   1
        3  2  1  −1
```

Note that zero entries are conveniently omitted: the relational representation of ordinary matrices is especially attractive when the matrices are "sparse", i.e., have a large proportion of zeros. The alternating tensor in particular, although a three-index array, not an ordinary matrix, is sparse: it has three 1s, three -1s and $3 \times 3 \times 3 - 6 =$ twenty-one 0s.
Relations are manipulated by accompanying operations of the *relational algebra* and of the *attribute algebra*. The former creates new relations from existing relations. The latter creates new attributes from existing attributes.
The relational algebra supports a *renaming assignment*, a *natural join* (a "binary" operation on two relations), and a *T-selector* (a "unary" operation on one relation).
Here are assignments to rename attributes in `t_dd` and `g_uu`.

```
g_uu'[i,joinon,vg <- i,j,v]g_uu;
t_dd'[joinon,j,vv <- i,j,v]t_dd;
```

The results are

```
g_uu'(i  joinon  vg)
      1    2      1
      2    1      1
      2    2     −2
```

```
t_dd'(joinon  j  vv)
        1     1   3
        1     2   1
        2     1  −1
```

The data are the same, but the attributes have been renamed. (For a subtle reason, this operation is considered part of the relational algebra, not of the attribute algebra.) Note that the prime character, `'`, is just part of the relation name in Aldat, not an operator as in MATLAB.
The *natural join* combines two relations on their common attribute(s) by merging all tuples from the first relation with all tuples from the second relation that match on values of that (those) shared attribute(s). Here is the result of joining the two relations we've just renamed so that they *have* a common attribute, `joinon`.

```
g_uu' natjoin t_dd'
(i  joinon  j  vg  vv)  vgv   v
 2     1    1   1    3    3   5
 2     1    2   1    1    1   1
 1     2    1   1   −1   −1  −1
 2     2    1  −2   −1    2   5
```

(Ignore for the moment the "virtual attributes" vgv and v, which will be defined when we reach the attribute algebra, below.)
Examine carefully how the (2, 1, 1) tuple from g_uu' is duplicated because two tuples from t_dd' share its joinon value of 1, and how the (2, 1, −1) tuple from t_dd' is duplicated because two tuples from g_uu' share its joinon value of 2. The above joined relation is displayed with the values of the common attribute, joinon, grouped to reveal this merge.
a) Show that 33 tuples result from the natural join of relation R(A,B), which has 4 tuples with B = 1, 5 tuples with B = 2, and 3 tuples with B = 3, with relation S(B,C), which has 3 tuples with B = 2, 6 tuples with B = 3 and 2 tuples with B = 4.
The *T-selector* combines *selecting* certain tuples from a relation according to some condition on the values of its attributes, with *projecting* specified attributes from the result. Here is an example which selects the value 1 from joinon and then projects on the attributes other than joinon.

```
[i,j,vg,vv] where joinon=1 in (g_uu' natjoin t_dd')
(i  j  vg  vv)
 2  1   1   3
 2  2   1   1
```

(The projection component must remove duplicates if any result from the disappearance of attribute(s).)
b) Show that only one tuple results from: [i] where joinon/abs(vg)=1 in g_uu'

The *attribute algebra* creates new attributes from old, independently of any relational context. So the results are *virtual attributes*, potentially available to any relation which has all the antecedent attributes. Thus

```
let vgv be vg*vv
```

defines the product of two attributes. Since these attributes are both in the relation which is the natural join above, g_uu' natjoin t_dd', the result, vgv, is meaningful in the context of this join and so can be shown in connection with it, as we did above. It is, however, not an attribute of this join, although it could be *actualized* in a new relation which is a projection of it.

```
gt <- [i,joinon,j,vg,vv,vgv] in (g_uu' natjoin t_dd')
```

```
gt(i  joinon  j  vg   vv  vgv)   v
   1     2    1   1   −1    −1  −1
   2     1    1   1    3     3   5
   2     2    1  −2   −1     2   5
   2     1    2   1    1     1   1
```

Again, ignore for the moment the virtual attribute v.
(Two small syntactical comments: <- is the *assignment* operator of the relational algebra which, without renaming attributes, creates a new named relation; omitting the where syntax gives pure projection, and a similar omission gives pure selection, as special cases of the

T-selector.)
Projection, and relational algebra expressions in general, may be used to actualize virtual attributes created by the attribute algebra.
c) Show that `vgv` is as shown in relation `gt` above.
The second operator of the attribute algebra that we must discuss is *aggregation*. This aggregates values within an attribute, creating a new attribute which is also virtual until actualized. For example, the virtual attribute `v` shown above beside `vgv` associated with the natural join `g_uu' natjoin t_dd'`, is the aggregate sum over tuples sharing common values of attributes `i`and `j`.

```
let v be equiv + of vv*vg by i,j
```

This groups tuples according to common values of `i` and `j` and sums the values of `vg*vv` within these groupings. The tuples in `gt` above are ordered to reveal these groups. Note that `vg*vv` is an unnamed virtual attribute. We could instead have used the name, `vgv`, we created above.
(We could invent syntax specialized for tensor index sums

```
let v be agg + of vg*vv on joinon
```

where the attributes `i` and `j` are the *complements* in, say, relation `gt`, of `vg, vv` and the `on` attribute `joinon`. But the syntax referring to equivalence classes of tuples is clearer to work with in the relational representation.)
If the Aldat programmer is careful in actualizing the virtual attribute `v` there will be no conflict with the attribute `v` which already actually appears in relations `g_uu` and `t_dd`. In a projection, or any relational algebra operation, virtual attribute definitions in the attribute algebra are ignored if an attribute of that name is already actual in any relation in that relational algebra expression.
d) Confirm that the three different values of v above are those that appear in the matrix product $g_{uu} \times t_{dd}$.
This choice of relational and attribute algebra operations permits us to multiply ordinary matrices. Here is the full code, starting with the original relational representations `g_uu(i,j,v)` and `t_dd(i,j,v)`.

```
t_dd'[joinon,j,vv <- i,j,v]t_dd;
g_uu'[joinon,i,vg <- i,j,v]g_uu;
let v be equiv + of vv*vg by i,j;
t_ud <- [i,j,v] in (t_dd' natjoin g_uu');
```

(Note that g_{uu} is symmetrical, so we may swap the first two attributes of its relational representation.)
e) Show that `t_ud` obtained above is the relational representation for the matrix t_{ud}.
f) Write Aldat code, analogous to the above, to calculate `t_du` and `t_uu`. Show that the latter can be derived in two different ways.
g) Represent `a_d, b_d, c_u, d_u` and `g_dd` from the previous Excursion as relations and write code to find `a_u, b_u, c_d, d_d`. (Calculating `g_dd` as the inverse of `g_uu` using, say, Gaussian elimination, can be done in Aldat, but there is no builtin operator to do this.)
The dot product, `a.b`, can also be coded relationally.

```
a_u'[joinon,vv <- i,v]a_u;
b_d'[joinon,vw <- i,v]b_d;
let v be red + of vv*vw;
ab <- [v] in (a_u' natjoin b_d');
```

The new attribute algebra operator, `red`, can be thought of as `equiv` followed by an empty `by`-list: there are no groupings and the sum is over the whole relation. (The syntax `let v be agg + of vv*vw on joinon` would also be valid here.)
h) Use `red +` to find the *trace* of, say, `t_ud`.
So far we have not advanced much beyond matrices and MATLAB. Now let's go to three

dimensions, with

$$g\_dd = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

(whose *determinant* $g = 1$) and the three-index *alternating tensor*

$$\frac{1}{\sqrt{g}}\epsilon^{ijk}$$

where $\epsilon^{ijk} = 1$ if (i, j, k) is an even permutation of (1,2,3), $\epsilon^{ijk} = -1$ if (i, j, k) is an odd permutation of (1,2,3), and otherwise zero. Here is the alternating tensor explicitly in Aldat

```
relation alt_uuu(i,j,k,v) <-
 {(1,2,3,1),(2,1,3,-1),(2,3,1,1),(3,2,1,-1),(3,1,2,1),(1,3,2,-1)};
```

(incidentally showing how to declare and how to initialize a relation). This was displayed at the beginning of this Excursion (with a different ordering of the tuples).
To lower the first index, we need the relational representation of `g_dd` and the Aldat code

```
alt_uuu'[joinon,j,k,va <- i,j,k,v]alt_uuu;
g_dd'[joinon,i,vg <- i,j,v]g_dd;
let v be equiv + of va*vg by i,j,k;
alt_duu <- [i,j,k,v] where v!=0 in (alt_uuu' natjoin g_dd');
```

giving

```
alt_duu(i  j  k  v)
        1  1  3  -1
        1  2  3   2
        1  3  1   1
        1  3  2  -2
        2  2  3   1
        2  3  2  -1
        3  1  2  -1
        3  2  1   1
```

i) Calculate the remaining six versions of `alt` and show (the following uses array, not relational, notation)

- $\texttt{alt}_{udu}(i, j, k) = \texttt{alt}_{duu}(k, i, j)$ and $\texttt{alt}_{uud}(i, j, k) = \texttt{alt}_{duu}(j, k, i)$;
- $\texttt{alt}_{dud}(i, j, k) = \texttt{alt}_{udd}(k, i, j)$ and $\texttt{alt}_{ddu}(i, j, k) = \texttt{alt}_{udd}(j, k, i)$;
- $\texttt{alt}_{duu}(i, j, k) = -\texttt{alt}_{duu}(i, k, j)$, $\texttt{alt}_{udu}(i, j, k) = -\texttt{alt}_{udu}(i, k, j)$ and $\texttt{alt}_{uud}(i, j, k) = -\texttt{alt}_{uud}(i, k, j)$;
- $\texttt{alt}_{udd}(i, j, k) = -\texttt{alt}_{udd}(i, k, j)$, $\texttt{alt}_{dud}(i, j, k) = -\texttt{alt}_{dud}(i, k, j)$ and $\texttt{alt}_{ddu}(i, j, k) = -\texttt{alt}_{ddu}(i, k, j)$; and
- $\texttt{alt}_{ddd} = \texttt{alt}_{uuu}$.

How many different ways are there to calculate, say, `alt_udd` from this starting point? `alt_ddd`?
j) Find a way to write out these alternating tensors as three-dimensional "matrices". Do this in a way to show the antisymmetries of each of the eight versions.
To run these programs other than by hand you must install a copy of Aldat on your computer: /citealdat.
Although Aldat is written in Java it is designed to run under UNIX-like operating systems. It is possible, if convoluted given the lack of polymorphism in nested relations, to write Aldat code analogous to `tensorCalc` in the previous Excursion.

35. Look up H. S. M. Coxeter's *Regular Polytopes* [Cox63] and use the interval algebra to construct higher-dimensional versions of the tetrahedron, cube and octahedron.

36. How might we use the interval algebra to describe a shear operation?

37. **Factoring sums and differences of squares by Clifford algebra.** Since the Clifford algebra gives a basis for a d-dimensional Euclidean space, we can use Pythagoras to find the square root of a sum of squares.

    2D $y^2 + z^2 = (ye_1 + ze_2)^2$ using the multiplication table

	y	z
y	1	e_{12}
z	$-e_{12}$	1

    3D $x^2 + y^2 + z^2 = (xe_1 + ye_2 + ze_3)^2$

    4D $w^2 + x^2 + y^2 + z^2 = (we_1 + xe_2 + ye_3 + ze_4)^2$

    We can also find the square root of differences of squares

    2D $y^2 - z^2 = (ye_1 + ze_{12})^2$ using the multiplication table

	y	z
y	1	e_2
z	$-e_2$	-1

    3D $x^2 + y^2 - z^2 = (xe_1 + ye_2 + ze_{12})^2$

    4D $w^2 + x^2 + y^2 - z^2 = (we_1 + xe_2 + ye_3 + ze_{123456})^2$

    but note that in 4D we had to go to a 6-dimensional space with a Clifford algebra of $2^6 = 64$ different elements because $e_{1..d}^2$ is -1 for $d = 2, 3, 6, 7, ..$ ($\text{sign}(e_{1..d}) = (-1)^{\lfloor d/2 \rfloor}$). Note, however, that $w^2 - x^2 - y^2 - z^2 = (we_1 + xe_{12} + ye_{13} + ze_{14})^2$ so $w^2 + x^2 + y^2 - z^2 = (iwe_{12} + ixe_{13} + iye_{14} + ize_1)^2$
    Alternatively we can find two *different* factors for a difference of squares

    2D $y^2 - z^2 = (y + z)(y - z)$

    3D $x^2 + y^2 - z^2 = (xe_1 + ye_2 + z)(xe_1 + ye_2 - z)$ using the multiplication table

	x	y	z
x	1	e_{12}	$-e_1$
y	$-e_{12}$	1	$-e_2$
z	e_1	e_2	-1

    4D $w^2 + x^2 + y^2 - z^2 = (we_1 + xe_2 + ye_3 + z)(we_1 + xe_2 + ye_3 - z)$

    Find a two-factor form ($\pm m$) for $E^2 - (p_x^2 + p_y^2 + p_z^2) - m^2$ (P.A.M.Dirac).

38. **Matrix representations of Clifford "numbers".** Ordinary numbers, even 2-numbers, are inadequate to represent the properties of Clifford numbers. Even in only two dimensions there is only 1 to give us all of $1^2 = 1$, $e_1^2 = 1$ and $e_2^2 = 1$, although there is i to give $e_{12}^2 = -1$. So we must turn to matrices.
    In two dimensions the complete Clifford algebra has four different members

	I		square is I
e_1		e_2	squares are I
	e_{12}		square is $-I$

We will need at least 2-by-2 matrices to provide enough choice to cover four independent matrices. For e_1 and e_2 we have a matrix and its inverse

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} \quad \text{and} \quad \frac{1}{ad-bc}\begin{pmatrix} d & -c \\ -b & a \end{pmatrix}$$

This is an embarrassment of riches. Let's suppose the determinant $ad - bc = 1$. So for the matrix to be its own inverse $a = d$ and $b = 0 = c$. That gives us only the identity I.
So now suppose the determinant $= -1$: $a = -d$ and $-a^2 - bc = -1$ or $a = \sqrt{1 - bc}$. We can set $bc = 0$ giving $a = 1$, or $a = 0$ giving $bc = 1$. So

$$\begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$$

are possible matrices. Show that they "anticommute", $e_1 * e_2 + e_2 * e_1 = 0$. Work out e_{12} and check that $e_{12}^2 = -I$.
(Matrices such as

$$\begin{pmatrix} 1 & \\ 3 & -1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} & 2 \\ 1/2 & \end{pmatrix}$$

also satisfy the self-inverse and det $= -1$ criteria. What is wrong with them?)
In three dimensions we have eight linearly independent elements

$$\begin{array}{ccc} & I & \\ e_1 & e_2 & e_3 \\ e_{12} & e_{13} & e_{23} \\ & e_{123} & \end{array} \qquad \begin{array}{l} \text{square is } I \\ \text{squares are } I \\ \text{squares are } -I \\ \text{square is } -I \end{array}$$

We can stretch 2-by-2 matrices to cover these if we allow 2-numbers, giving us room for 2*2-by-2 independent parameters. In the above discussion of two dimensions with det $= -1$ we can now also have $b = i = -c$. If we take the three matrices

$$\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}, \quad \begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} & -i \\ i & \end{pmatrix}$$

as e_1, e_2 and e_3 respectively, we can find the "dot product" of these three with the components x, y, z of $\vec{r}$

$$\vec{r} \cdot \vec{e} = \begin{pmatrix} x & y - iz \\ y + iz & -x \end{pmatrix}$$

(The "Pauli matrices" (see Week 6 Note 5, Book 8c Part III Note 27, Book 8c Part IV Note 32) are formulated to give the above dot product as

$$\vec{r} \cdot \vec{e} = \begin{pmatrix} z & x - iy \\ x + iy & -z \end{pmatrix}$$

Find the three Pauli matrices and say how they relate to e_1, e_2 and e_3 above. Show that the squares of dot products with $\vec{r}$ are $r^2 I$.)
In four dimensions the sixteen elements force us beyond 2-by-2 matrices.

$$\begin{array}{cccccc} & & I & & & \\ & e_1 & e_2 & e_3 & e_4 & \\ e_{12} & e_{13} & e_{14} & e_{23} & e_{24} & e_{34} \\ & e_{123} & e_{124} & e_{134} & e_{234} & \\ & & e_{1234} & & & \end{array} \qquad \begin{array}{l} \text{square is } I \\ \text{squares are } I \\ \text{squares are } -I \\ \text{squares are } -I \\ \text{square is } I \end{array}$$

Three-by-3 matrices using 2-numbers could give us these 16 elements, but 4-by-4 matrices using 2-numbers and symmetric in the first of the pair and anti-symmetric in the second component give us exactly 16: 10 for the symmetric part and 6 for the antisymmetric part. (Such matrices are "Hermitian" and important in quantum physics: see Excursion *Matrix Mechanics* below.)
A somewhat different sixteen are given by an easy way to step from the 2-by-2 matrices for three dimensions to these 4-by-4 matrices. Here are the four for e_1, e_2, e_3 and e_4 (the last three are lumped together for $j = 1, 2, 3$).

$$\begin{pmatrix} I & \\ & -I \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} & -i * e_j \\ i * e_j^\dagger & \end{pmatrix}$$

where each of these is a 4-by-4 matrix described using 2-by-2 blocks and $e_j^\dagger$ is e_j transposed and with the sign of its second 2-number component reversed.
(These are related to the "Dirac matrices" (Week 7a Note 16) which serve a different but related role.

$$\begin{pmatrix} I & \\ & -I \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} & e_j \\ -e_j^\dagger & \end{pmatrix}$$

See the last part of the previous Excursion.)

39. **Matrix Mechanics.** a) The zero anticommutator for Clifford numbers means that the sign of a product changes when the two terms are exchanged

$$[e_1, e_2]_+ = e_1e_2 + e_2e_1 = 0 \Leftrightarrow e_2e_1 = -e_1e_2$$

This reminds us of fermions, electrons in particular: see Week 5 Note 9.
So we might wonder if we can use Clifford numbers to represent quantities in quantum mechanics. Let's focus on electrons.
Electrons, considered in isolation, each have two states which we call "spin-up" and "spin-down". This is because they have spin 1/2 and that is because turning an electron 360 degrees is the same relativistically as exchanging two electrons (see Excursion *Spin Statistics, etc.* in Week 7a). Week 6 explores matrices which rotate such two-state systems and introduces the Pauli matrices (Note 3) as a convenient description.
Here we consider the Pauli-matrix representation of Clifford algebra more fundamentally, as actually representing quantities describing two-state systems such as electrons.
Each such quantity will generally have two values, which must be regular, 1-dimensional numbers. (In the rest of this Excursion I'll be conventional and call such numbers "real", 2-dimensional numbers "complex" and the i part of a complex number "imaginary".)
The two numbers associated with a 2-by-2 matrix are its eigenvalues (Week 3 Note 5). The two eigenvalues λ_1 and λ_2 are especially easy to find in a 2-by-2 matrix M. We need only the determinant and the trace.

$$\begin{aligned} \mathrm{det}M &= \lambda_1 \times \lambda_2 \\ \mathrm{tr}M &= \lambda_1 + \lambda_2 \end{aligned}$$

So for the Pauli matrices

$$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \quad \begin{pmatrix} & -i \\ i & \end{pmatrix} \quad \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$$

which each have determinant -1 and trace 0, the eigenvalues must be 1 and -1. This is very suitable for the two states of spin. (We could multiply by 1/2 or by $\hbar/2$ to make it all more physical but we choose to keep it simple.)

So we have two different concepts: a quantity, which is represented by a matrix; and its values, which are the set of eigenvalues of the matrix. Note that this way of viewing a quantity includes *all* its values at once. This is a profound difference between quantum mechanics and pre-quantum physics.
b) *Hermitian matrices.* If the quantity is represented by a hermitian matrix it will have real values. A hermitian matrix equals its own complex transpose ("Hermitian conjugate").

$$H^\dagger = H^{*T} = H \qquad \begin{pmatrix} w+z & x-iy \\ x+iy & w-z \end{pmatrix}^\dagger = \begin{pmatrix} w+z & x+iy \\ x-iy & w-z \end{pmatrix}^T = \begin{pmatrix} w+z & x-iy \\ x+iy & w-z \end{pmatrix}$$

where * is the complex conjugate operator, $(x+iy)^* = x - iy$, and T is the ordinary matrix transpose operator.
Hermitian quantities have real values.

$$\begin{aligned} \det \begin{pmatrix} w+z & x-iy \\ x+iy & w-z \end{pmatrix} &= (w+z)(w-z) + (x+iy)(x-iy) = w^2 - (x^2+y^2+z^2) \\ \mathrm{tr} \begin{pmatrix} w+z & x-iy \\ x+iy & w-z \end{pmatrix} &= (w+z) + (w-z) = 2w \end{aligned}$$

"AntiHermitian" matrices are the negative of their Hermitian conjugate. We can generate any antiHermitian matrix by multiplying the corresponding Hermitian matrix by i and vice-versa.
The commutator of two Hermitian matrices is antiHermitian.

$$[H_1, H_2]^\dagger = (H_1H_2 - H_2H_1)^\dagger = H_2^\dagger H_1^\dagger - H_1^\dagger H_2^\dagger = H_2H_1 - H_1H_2 = -[H_1, H_2]$$

because the ordinary transpose of a product is the product of the transposes in reverse order. (Show this.) (What about the commutator of two antiHermitian matrices? Of a Hermitian and an antiHermitian matrix? What about the anticommutators in all these cases?)
We will represent all physical quantities by Hermitian matrices.
The square of a Hermitian matrix is non-negative. That is, its values are non-negative. We can see this by using the matrix of eigenvectors, U, to transform the Hermitian matrix H to its diagonal form D, and noting that the inverse of U is just $U^\dagger$ for a U which diagonalizes a Hermitian matrix (just as rotations diagonalize a symmetric matrix and the inverse of a rotation matrix is its transpose).

$$D^2 = U^\dagger H^2 U = U^\dagger H U U^\dagger H U = DD$$

So the square of a diagonal matrix is diagonal and its elements are just the squares of the corresponding elements of the original diagonal matrix. Thus all its values are non-negative and so by definition we can call the matrix non-negative. (We'll write $H^2 \geq 0$ to express this even though the notation combines matrices and numbers in a non-literal way.)
c) *Diagonalization.* Here is this diagonalization transformation for the two Pauli matrices that are not already diagonal.

matrix	eigenvectors		matrix	eigenvectors	
$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$\begin{pmatrix} & -i \\ i & \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix}$

so we can write, combining the pairs of eigenvectors into matrices,

$$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} & -i \\ i & \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ i & i \end{pmatrix}$$

and finally we can get the diagonalized matrices by using the Hermitian conjugates as the inverses

$$\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$$

and

$$\frac{1}{2}\begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}\begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix} = \begin{pmatrix} & -i \\ i & \end{pmatrix}$$

What we must notice is that the diagonalization transformations are not the same for the two matrices. The two matrices are *not simultaneously diagonalizable.* This must be interpreted to mean that we cannot simultaneously measure the quantities represented by the two matrices. In the state in which the first is diagonal, and therefore can be measured to give the value 1 or -1, the second is transformed to

$$\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} & -i \\ i & \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} & i \\ -i & \end{pmatrix}$$

Quantities are simultaneously measurable if their matrices are simultaneously diagonalizable. Matrices are simultaneously diagonalizable iff they commute.
Only if: diagonalize both matrices; diagonal matrices commute.
If: Suppose one matrix is diagonal but the other has a single off-diagonal element e. Then the commutators will contain components

$$\begin{pmatrix} a & \\ & b \end{pmatrix}\begin{pmatrix} c & e \\ & d \end{pmatrix} - \begin{pmatrix} c & e \\ & d \end{pmatrix}\begin{pmatrix} a & \\ & b \end{pmatrix} = \begin{pmatrix} ac & ae \\ & bd \end{pmatrix} - \begin{pmatrix} ac & be \\ & bd \end{pmatrix}$$

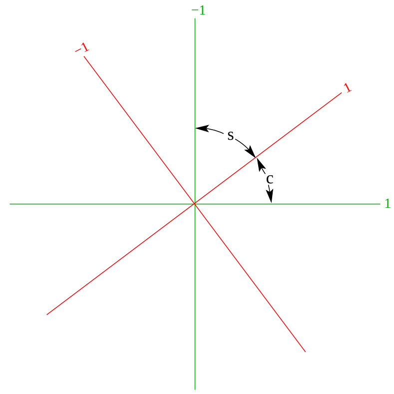
For this to vanish e must be 0.
So the commutator plays a central role in simultaneous measurability.
d) *Expected values.* Heisenberg expressed the impossibility of exact simultaneous measurements as a joint *uncertainty* in the results of the two measurements. He found

$$< (A- < A >)^2 >< (B- < B >)^2 > \quad \geq \quad |\frac{1}{2} < [A, B] >|^2$$

where $< K >$ is the expected value of the quantity K and so $< (A- < A >)^2 >$ is the *variance* of the quantity. And there's the commutator again.
The expected value of a quantity described by a 2-by-2 matrix is just the two possible values each weighted by the probability of it occurring. The values are the eigenvalues. How do we find their probabilities? These depend on the *state* the system is in when we make the measurement. Since there are two probabilities, they must form a vector in the same 2-D space the matrix can be thought of as operating in. In fact, there will be two related vectors, each being a *probability amplitude* (see Week 5), so that their dot product has two components, being the two probability amplitudes, whose squares sum to 1. That is, the vectors, which may be complex, are normalized.
It may be best to draw a picture to explain this, and I've chosen another variant of the same two eigenvalues, 1 and -1, transformed into a purely real matrix, to make the picture possible.

$$\begin{pmatrix} 4/5 & 3/5 \\ 3/5 & -4/5 \end{pmatrix}\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}\begin{pmatrix} 4/5 & 3/5 \\ 3/5 & -4/5 \end{pmatrix} = \frac{1}{25}\begin{pmatrix} 7 & 24 \\ 24 & -7 \end{pmatrix}$$

(where the transformation matrix coincidentally and handily happens to be its own inverse). We'll call the original matrix the green matrix. Its eigenvectors are $(1, 0)^T$ and $(0, 1)^T$. Let's suppose there is a red matrix whose eigenvectors are $(4/5, 3/5)^T$ and $(3/5, -4/5)^T$, also with eigenvalues 1 and -1 as shown.



Now suppose that the system is in state $(4/5, 3/5)^T$: the red quantity is diagonal and has the value $+1$. The green value $+1$ is separated from this state by an angle whose cosine is c and the green value -1 is separated from it by an angle whose cosine is s. Since $c^2 + s^2 = 1$ (for these complementary angles) the two probabilities can only be c^2 and s^2. The expected value is

$$1 \times c^2 + (-1) \times s^2 = c^2 - s^2 = (4/5)^2 - (3/5)^2 = 7/25$$

Note that this is the same as

$$(4/5, 3/5)\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}\begin{pmatrix} 4/5 \\ 3/5 \end{pmatrix} = (1,0)\frac{1}{25}\begin{pmatrix} 7 & 24 \\ 24 & -7 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

where the latter expression gives the same result from the point of view of the red quantity. These results both give the expected value of the green quantity in the case that the red quantity has been measured to have the exact value 1.

The expected value of a matrix M given that we're in state σ is thus $< M >= \sigma^\dagger M \sigma$. Show that $< M + N >=< M > + < N >$ and $< aM >= a < M >$ for matrices M and N and number a. Note that $< MN > \neq < M >< N >$ in general: find a counterexample.

e) *Uncertainty.* Now for Heisenberg's uncertainty principle.

$$< (A - < A >)^2 >< (B - < B >)^2 > \quad \geq \quad |\frac{1}{2} < [A, B] >|^2$$

Because $< A >$ and $< B >$ are just numbers they make no difference in the commutator, so Heisenberg's result follows if

$$< K^2 >< L^2 > \quad \geq \quad |\frac{1}{2} < [K, L] >|^2$$

for Hermitian matrices K and L.

This inequality follows, as we'll see, from

$$| x + iy |^2 = x^2 + y^2 \quad \geq \quad y^2 = | iy |^2$$

for numbers x and y, and also from

$$(H_1 + iH_2)(H_1 - iH_2) \geq 0$$

for any two Hermitian matrices H_1 and H_2, i.e., that the eigenvalues of $(H_1 + iH_2)(H_1 - iH_2)$ are non-negative. Show these results.

In this latter we'll let $H_1 = K + uL$ and $H_2 = vL$ so

$$\begin{aligned} 0 &\leq (H_1 + iH_2)(H_1 - iH_2) \\ &= (K + uL + ivL)(K + uL - ivL) \\ &= K^2 + (u + iv)(u - iv)L^2 + (u - iv)KL + (u + iv)LK \end{aligned}$$

For the next step, we'll let $u + iv = - < KL > / < L^2 >$. Since $0 \leq M$ means that the eigenvalues of M are non-negative it follows that the expectation value of this M is also non-negative, $0 \quad \leq \ < M >$.

$$\begin{aligned}
0 &\leq < K^2 + (u+iv)(u-iv)L^2 + (u-iv)KL + (u+iv)LK > \\
&= < K^2 > +(u-iv)(u+iv) < L^2 > +(u-iv) < KL > +(u+iv) < LK > \\
&= < K^2 > -(u-iv) < KL > +(u-iv) < KL > - < KL >< LK > / < L^2 > \\
&= < K^2 > - \frac{< KL >< LK >}{< L^2 >}
\end{aligned}$$

If $< L^2 > \neq 0$ it follows that $< KL >= 0$ ([Jor86, p.131]: I've extracted most of the discussion of this Excursion from this valuable little book) and so $< K^2 >< L^2 >= 0 =< KL >< LK >$.
So $< K^2 >< L^2 > \ \geq \ < KL >< LK >$ for Hermitian matrices K and L.
You can show that $< LK > = < KL >^*$, the complex conjugate of $< KL >$.
So $< K^2 >< L^2 > \ \geq \ < KL >< KL >^*$ for Hermitian matrices K and L.
Now

$$\begin{aligned}
KL &= \frac{1}{2}(KL + LK) - \frac{1}{2}(Kl - LK) \\
&= \frac{1}{2}[K, L]_+ + i\frac{1}{2}(-i)[K, L]_- \\
&= H_1 + iH_2
\end{aligned}$$

where $[,]_+$ marks the anticommutator, which is Hermitian if its operands are, and $[,]_-$ marks the commutator, which is antiHermitian if its operands are Hermitian. So $(-i)[,]_-$ is Hermitian in this case.
Thus, finally,

$$\begin{aligned}
< K^2 >< L^2 > &\geq < KL >< KL >^* \\
&\geq < H_1 + iH_2 >< H_1 - iH_2 > \\
&= (< H_1 > +i < H_2 >)(< H_1 > -i < H_2 >) \\
&= |< H_1 > +i < H_2 >|^2 \\
&\geq |\, i < H_2 >|^2 \\
&= |\, i\frac{1}{2}(-i)[K, L]_- \,|^2 \\
&= |\, \frac{1}{2}[K, L]_- \,|^2
\end{aligned}$$

This is what we needed, at the start of this discussion of *Uncertainty*, to show Heisenberg's uncertainty principle

$$< (A- < A >)^2 >< (B- < B >)^2 > \ \geq \ |\, \frac{1}{2} < [A, B]_- >|^2$$

f) *Uncertainty examples: Pauli matrices.* Show that the expectation values for each of the three Pauli matrices are 1 and -1 respectively for the two state vectors for which the matrix is diagonal and 0 for the state vectors for which the other Pauli matrices are diagonal. Here's a start

$$\begin{aligned}
\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} &= \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \\
\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}\begin{pmatrix} & -i \\ i & \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} &= \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}
\end{aligned}$$

So $\frac{1}{\sqrt{2}}(1,1)^T$ is the state vector that gives 1 as the expected value of $\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$ and $\frac{1}{\sqrt{2}}(1,-1)^T$ gives -1 as its expected value; $\frac{1}{\sqrt{2}}(1,-i)^\dagger$ gives 1 as the expected value of $\begin{pmatrix} & -i \\ i & \end{pmatrix}$ and $\frac{1}{\sqrt{2}}(1,i)^\dagger$ gives -1 as its expected value. (Note that the latter vectors must be complex-conjugated when they are transposed.)
Here are the expectation values you should get for the three Pauli matrices for the various states.

	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$	1	-1	0	0	0	0
$\begin{pmatrix} & -i \\ i & \end{pmatrix}$	0	0	1	-1	0	0
$\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$	0	0	0	0	1	-1

Now show that $<(A-<A>I)^2>$ is either the identity matrix I or one of

$$\begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 2i \\ -2i & 2 \end{pmatrix}, \begin{pmatrix} 2 & -2i \\ 2i & 2 \end{pmatrix}, \begin{pmatrix} & \\ & -2 \end{pmatrix}, \begin{pmatrix} & \\ & 2 \end{pmatrix}$$

depending on the Pauli matrix and the state.
The three commutators are

$$\begin{aligned}
\left[\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\begin{pmatrix} & -i \\ i & \end{pmatrix}\right]_- &= \begin{pmatrix} 2i & \\ & -21 \end{pmatrix} \\
\left[\begin{pmatrix} & 1 \\ 1 & \end{pmatrix}\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}\right]_- &= \begin{pmatrix} & -2 \\ 2 & \end{pmatrix} \\
\left[\begin{pmatrix} & -i \\ i & \end{pmatrix}\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}\right]_- &= \begin{pmatrix} & 2i \\ 21 & \end{pmatrix}
\end{aligned}$$

Show that the expectation values for these commutators are either 0 or $\pm 2i$ depending on the state and Pauli matrix, for the above states.
Put the results together and show that the Heisenberg uncertainty relations are satisfied with exact equality (either $0=0$ or $1=1$) each time.
A slightly more interesting exact equality is given by the state $(4/5,3/5)^T$: try it.

40. **Spin and Clifford Algebra.**
a) *One spin.* In 3D space a spin must have a direction, which we can suppose to be given by three coordinates x,y,z with $x^2+y^2+z^2=1$. These are not coefficients of a vector because a spin is not a vector. (If we look at a spin in a mirror its direction appears reversed.)
We introduced Clifford algebra to deal with rotations, so it is plausible to describe the spin as

$$xe_1 + ye_2 + ze_3$$

For spin 1/2, which has two possible values (we'll call them 1 and -1) the e_j must be 2-by-2 matrices, the Pauli matrices.
The noncommutativity of the e_j, and hence the fact that they cannot be diagonalized simultaneously, reveals that we cannot simultaneously measure the x,y or z values of the spin.
In fact the *Uncertainty examples: Pauli matrices* part of Excursion *Matrix Mechanics* shows that if one Pauli matrix has a definite value of 1 or -1 then the other two Pauli matrices are

completely undetermined: they have expected values 0 which shows that they will measure 1 or -1 with equal probability, 1/2.
Interpreting the three Pauli matrices as the basis for a rotational space in three dimensions, this tells us that we can determine the spin in any one direction but only at the expense of finding it completely undetermined in the other two perpendicular directions.
We can use the components x, y and z to describe an arbitrary direction

$$xe_1 + ye_2 + ze_3 = \begin{pmatrix} z & x - iy \\ x + iy & -z \end{pmatrix}$$

This diagonalizes to

$$\begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$$

just as e_1, e_2 and e_3 each do.
(It is particularly easy to find the eigenvalues λ for a 2-by-2 matrix. The "characteristic equation" for λ is

$$\lambda^2 - \text{trace} * \lambda + \det = 0$$

because

$$0 = \det \begin{pmatrix} a - \lambda & c \\ b & d - \lambda \end{pmatrix}$$

So for

$$\begin{pmatrix} z & x - iy \\ x + iy & -z \end{pmatrix}$$

trace $= 0$ and det $= x^2 + y^2 + z^2 = 1$: $\lambda^2 = 1$ so $\lambda = \pm 1$.)
We can always re-assign coordinates so that the direction we're measuring is the z-direction, so, for one spin, this bit of algebra is not needed. But we'll need it more when we come to two spins.
Note that we can make three successive measurements, say, first in the z-direction and discard all particles whose spin is not $+1$ (i.e., those of spin -1), then, second, measure the particles we kept along the x-direction and discard all particles not having some particular spin (say 1: it doesn't matter just now), then, third, in the z-direction again.
You might think that *all* particles that come out of the third measurement would have spin $+1$, because we already selected for them in the first measurment. Not so. The third measurement will again have equal z-probability of $+1$ or -1 as a result of the second measurement.
That second measurement should remind you of inserting a polarizing filter between two others at right angles to each other, as we did in Week 1. Light gets through the last filter where it didn't before. This quantum phenomenon of spin is analogous.
(The above split-beam experiment was first done with silver atoms—where valence electrons in the atoms caused the effect—in a magnetic field by Gerlach and Stern in 1922.)
b) *Two spins.* Now consider a system of two particles each with spin 1/2. The matrices must double in size to handle this double load.
We're still in three dimensions, so each particle must be described by three matrices and corresponding coefficients, one matrix for each basis direction in space, and the three matrices must satisfy the Clifford algebra rules of squaring to I and of anticommuting. In addition they will satisfy the Pauli conditions $e_1e_2 = ie_3$, etc.
But the two particles are separate and their spins can be measured simultaneously. So the matrices describing one particle must all commute with all of the matrices describing the other,
The way to extend the 2-by-2 Pauli matrices to 4-by-4 is by *tensor product* (or "Kronecker product" in MATLAB).
Using the Pauli matrices, e_1, e_2, e_2, here are the two sets.

$$I \times e_1, \quad I \times e_2, \quad I \times e_3$$

$$e_1 \times I, \quad e_2 \times I, \quad e_3 \times I$$

where

$$A \times B \stackrel{\text{def}}{=} \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix}$$

in the 2-by-2 case: this is a 4-by-4 matrix shown as 2-by-2 blocks,
Thus

$$I \times e_1 = \begin{pmatrix} & 1 & & \\ 1 & & & \\ & & & 1 \\ & & 1 & \end{pmatrix} \qquad e_1 \times I = \begin{pmatrix} & & 1 & \\ & & & 1 \\ 1 & & & \\ & 1 & & \end{pmatrix}$$

$$I \times e_2 = \begin{pmatrix} & -i & & \\ i & & & \\ & & & -i \\ & & i & \end{pmatrix} \qquad e_2 \times I = \begin{pmatrix} & & -i & \\ & & & -i \\ i & & & \\ & i & & \end{pmatrix}$$

$$I \times e_3 = \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix} \qquad e_3 \times I = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}$$

(You should now be able to formulate three or more spins as combined Clifford algebras: try combining three or more 2-by-2 matrices with tensor products, doubling the size of the matrix each time.)
Some rules for tensor products are useful. You can show that $(A \times B)(C \times D) = AC \times BD$ where the 4-by-4 matrices in parentheses are combined by ordinary matrix multiplication.
You can show that $A \times I + B \times I = (A + B) \times I$ and vice-versa (with the I first).
And you can show that $A \times B \neq B \times A$ in general.
With these, you can show that each set above forms a Clifford algebra, e.g.,

$$(I \times e_j)^2 = I \times I, \quad [I \times e_j, I \times e_k]_+ = 0,$$

that each set satisfies Pauli's $(I \times e_1)(I \times e_2) = i(I \times e_3)$, etc., and that the two sets commute with each other

$$[I \times e_j, e_k \times I]_- = e_k \times e_j - e_k \times e_j = 0$$

It is also easy to diagonalize these tensor products of Pauli matrices. Suppose the eigenvectors are given by the columns of S_j for the 2-by-2 Pauli matrix e_j. Thus $S_j^\dagger e_j S_j = d$ with $d = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$.
Then we could try $I \times S_j$ to diagonalize the first set and $S_j \times I$ for the second.
But then $I \times e_j$ and $e_j \times I$ are not diagonalized by the same set of eigenvectors: they are not simultaneously diagonalized. So instead we use $S_j \times S_j$ for both.
So

$$(S_j^\dagger \times S_j^\dagger)(I \times e_j)(S_j \times S_j) = S_j^\dagger S_j \times S_j^\dagger e_j S_j = I \times d$$

and

$$(S_j^\dagger \times S_j^\dagger)(e_j \times I)(S_j \times S_j) = S_j^\dagger e_j S_j \times S_j^\dagger S_j = d \times I$$

We've worked out the S_j in part *Uncertainty examples: Pauli matrices* of Excursion *Matrix Mechanics*. From them we get

$$\begin{array}{ccc} S_1 & S_2 & S_3 \\ \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} & \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ i & -i & i & -i \\ i & i & -i & -i \\ -1 & 1 & 1 & -1 \end{pmatrix} & \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \end{array}$$

To describe the spins in each direction we combine the spin matrices by adding them

$$\begin{array}{ccc} I\times e_1+e_1\times I & I\times e_2+e_2\times I & I\times e_3+e_3\times I \\ \begin{pmatrix} & 1 & 1 & \\ 1 & & & 1 \\ 1 & & & 1 \\ & 1 & 1 & \end{pmatrix} & \begin{pmatrix} & -i & -i & \\ i & & & -i \\ i & & & -i \\ & i & i & \end{pmatrix} & \begin{pmatrix} 2 & & & \\ & & & \\ & & & \\ & & & -2 \end{pmatrix} \end{array}$$

And it's useful to have the squares, too.

$$\begin{pmatrix} 2 & & & 2 \\ & 2 & 2 & \\ & 2 & 2 & \\ 2 & & & 2 \end{pmatrix} \begin{pmatrix} 2 & & & -2 \\ & 2 & 2 & \\ & 2 & 2 & \\ -2 & & & 2 \end{pmatrix} \begin{pmatrix} 4 & & & \\ & & & \\ & & & \\ & & & 4 \end{pmatrix}$$

When we diagonalize any of these directions, to make a measurement, we get

$$\begin{pmatrix} 2 & & & \\ & 0 & & \\ & & 0 & \\ & & & -2 \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix} + \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}$$

Thus the measured spins are $2 = 1{+}1$, $0 = (-1){+}1$, $0 = 1{+}(-1)$ and $-2 = (-1) + (-1)$. (Remember, we must divide everything by 2 since I use 1 to describe spin 1/2.)
Now we look at the expected values for the various states in which each direction is diagonalized.

$$\frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} & 1 & 1 & \\ 1 & & & 1 \\ 1 & & & 1 \\ & 1 & 1 & \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & - & - & - \\ - & 0 & - & - \\ - & - & 0 & - \\ - & - & - & -2 \end{pmatrix}$$

(This is shorthand for

$$(1\ \ 1\ \ 1\ \ 1)\,()\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad (\quad 1\ \ -1\ \ 1\ \ -1)\,()\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

etc., so we're interested only in the diagonal elements of the result.)
You can show the same result for $I \times e_2 + e_2 \times I$ and $I \times e_3 + e_3 \times I$.
We'll now focus on states with spin 0 because spin-0 systems raised some deep questions for Einstein, Podolsky and Rosen in 1935. That is, we'll look at states such as the middle ones $(1\ \ -1\ \ 1\ \ -1)$ or $(1\ \ 1\ \ -1\ \ -1)$ for which

$$< I \times e_1 + e_1 \times I >= 0$$

and also (you should show this from the matrix above)

$$< (I \times e_1 + e_1 \times I)^2 >= 0$$

We are going to want to show that the expected value of measuring the first spin in direction (x, y, z) together with the second spin in direction (x', y', z') is

$$< (x(I\times e_1) + y(I\times e_2) + z(I\times e_3))(x'(e_1\times I) + y'(e_2\times I) + z'(e_3\times I)) >= -xx' - yy' - zz'$$

For this we must show that the combination

$$\begin{array}{cccc} & x & y & z \\ x' & < e_1 \times e_1 > & < e_1 \times e_2 > & < e_1 \times e_3 > \\ y' & < e_2 \times e_1 > & < e_2 \times e_2 > & < e_2 \times e_3 > \\ z' & < e_3 \times e_1 > & < e_3 \times e_2 > & < e_3 \times e_3 > \end{array} = \begin{array}{ccc} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{array}$$

for states with total spin $= 0$.
We can get the -1s from the expected values of the squares.

$$\begin{aligned} 0 &= < (I \times e_1 + e_1 \times I)^2 > \\ &= < (I \times e_1)^2 + 2(I \times e_1)(e_1 \times I) + (e_1 \times I)^2 > \\ &= < I \times I > +2 < e_1 \times e_1 > + < I \times I > \\ &= 2(1+ < e_1 \times e_1 >) \end{aligned}$$

So $< e_1 \times e_1 >= -1$.
The corresponding squares for e_2 and e_3 give the other two diagonal -1s.
Note that this part of the argument depends on the total spin being 0.
For the 0s we need a preliminary result. Consider two matrices, U and V, which are Clifford elements

$$U^2 = I = V^2 \qquad \text{and} \qquad [U, V]_+ = 0$$

Then, for any state

$$< U >^2 + < V >^2 \leq 1$$

Proof. For any two numbers p and q

$$(pU + qV)^2 = (p^2 + q^2)I \stackrel{\text{def}}{=} s^2 I$$

So the only value of this square is s^2. Hence the possible values of $pU + qV$ are $-s$ and s, and the expected value lies between them

$$-s \quad \leq \quad < pU + qV >= p < U > +q < V > \quad \leq \quad s$$

Now a trick: p and q can be any numbers, so let them be $< U >$ and $< V >$ respectively.

$$\begin{aligned} s \quad \geq \quad p < U > +q < V > &= < U >^2 + < V >^2 \\ &= p^2 + q^2 \\ &= s^2 \end{aligned}$$

So $s \geq s^2$ which means $s \leq 1$ and

$$< U >^2 + < V >^2 \quad \leq \quad s \leq 1 \qquad \square$$

Now we can put

$$\begin{aligned} U &= (I \times e_1)(e_1 \times I) = e_1 \times e_1 \\ V &= (I \times e_2)(e_1 \times I) = e_1 \times e_2 \end{aligned}$$

and since

$$(I \times e_j)^2 = I \times I$$

(which is just the 4-by-4 identity matrix I) so

$$\begin{aligned} ((I \times e_1)(e_1 \times I))^2 &= (I \times e_1)^2 (e_1 \times I)^2 = I \times I \\ ((I \times e_2)(e_1 \times I))^2 &= (I \times e_2)^2 (e_1 \times I)^2 = I \times I \end{aligned}$$

Also

$$
\begin{aligned}
& (I\times e_1)(e_1\times I)(I\times e_2)(e_1\times I)+(I\times e_2)(e_1\times I)(I\times e_1)(e_1\times I) \\
= \; & (I\times e_1)(I\times e_2)(e_1\times I)^2+(I\times e_2)(I\times e_1)(e_1\times I)^2 \\
= \; & (I\times e_1)(I\times e_2)+(I\times e_2)(I\times e_1) \\
= \; & 0
\end{aligned}
$$

So this U and V satisfy the premises of the above preliminary result and

$$1 \quad \geq \quad < e_1\times e_1 >^2 + < e_1\times e_2 >^2 = 1 + < e_1\times e_2 >^2$$

making $< e_1\times e_2 >= 0$.
This argument repeats for all the other $< e_j\times e_k >$, $j\neq k$ in the case of total spin 0.
So we have the result that the expected value of the product of the two spins, if the total spin is zero, measured in two different directions x,y,z and x',y',z' is

$$-xx'-yy'-zz'$$

From this result we can go on to get the probabilities of measuring the four different spin combinations in the two directions.
But first I should perhaps say why I've taken the above somewhat abstract and general approach rather than just calculating the expectation values directly from various state vectors. For one thing, there are many state vectors. The beginning of the discussion oversimplified by selecting specific state vectors to get results such as $< I\times e_1+e_1\times I >= 0$. So the discussion is not general (although the results are: see [Jor86, p.104]).
Second, finding specific states would require finding eigenvectors even more complicated than for

$$\begin{pmatrix} z & x-iy \\ x+iy & -z \end{pmatrix}$$

in the *One spin* part above. The two-spin matrix is

$$\begin{pmatrix} z+z' & x-iy & x'-iy' & \\ x+iy & -z+z' & & x'-iy' \\ x'+iy' & & z-z' & x-iy \\ & x'+iy' & x-iy & -z-z' \end{pmatrix}$$

I don't know how to do this neatly.
c) *Probabilities.* We showed in part (b) that the expected value of the *product* of the two measurements is $-xx'-yy'-zz'$. I'll abbreviate this as $-\vec{r}\cdot\vec{r'}$ where $\vec{r}=(x,y,z)$ and $\vec{r'}=(x',y',z')$ (and where we remember $\vec{r}\cdot\vec{r}=r^2=1$ and $\vec{r'}\cdot\vec{r'}=r'^2=1$.
We get a "magic square" of four possible results of the measurement, which I have surrounded with the expected values of all possible pairs.

$$
\begin{array}{c|cc|cc|c}
\cline{2-5}
 & 1 & 1 & 1 & -1 & \vec{r}\cdot(I\times\vec{e}) \\
\cline{2-5}
 & -1 & 1 & -1 & -1 & \vec{r}\cdot(I\times\vec{e}) \\
\cline{2-5}
\multicolumn{1}{c}{-\vec{r}\cdot\vec{r'}} & \multicolumn{2}{c}{\vec{r'}\cdot(\vec{e}\times I)} & \multicolumn{2}{c}{\vec{r'}\cdot(\vec{e}\times I)} & -\vec{r}\cdot\vec{r'}
\end{array}
$$

That is, the diagonal pairs 1,1 and $-1,-1$ both have product 1 while the off-diagonal pairs $1,-1$ and $-1,-1$ both have product -1: the expected value of this product we found to be $-\vec{r}\cdot\vec{r'}$.
The combination of the pair 1,1 and $1,-1$ in the top row has the first measurement 1 while the bottom row pair, $-1,1$ and $-1,-1$ has the first measurement -1: the expected value of

the first measurement is $x < I \times e_1) > +y < I \times e_2 > +z < I \times e_3 >$.
The combination of the pair 1,1 and $-1, 1$ in the top column has the second measurement 1 while the bottom column pair, $1, -1$ and $-1, -1$ has the second measurement -1: the expected value of the second measurement is $x' < e_1 \times I) > +y' < e_2 \times I > +z' < e_3 \times I >$.
We can turn expectated values into probabilities, in these cases with only the two values 1 and -1, by using the two equations

$$\begin{aligned} p_1 + p_{-1} &= 1 \\ p_1 - p_{-1} &= \text{expectation} \end{aligned}$$

So

$$\begin{aligned} p_1 &= (1 + \text{expectation})/2 \\ p_{-1} &= (1 - \text{expectation})/2 \end{aligned}$$

Thus the probability of the diagonal is

$$p(1,1) + p(-1,-1) = (1 - \vec{r} \cdot \vec{r'})/2$$

and the probability of the off-diagonal is

$$p(1,-1) + p(-1,1) = (1 + \vec{r} \cdot \vec{r'})/2$$

For the rows and columns we go back to the "preliminary result" of part (b) to show that the other two expected values are 0 and hence each row has equal probability 1/2 and each column has equal probability 1/2.
We set $U = (I \times e_1)(e_1 \times I) = e_1 \times e_1$ so $U^2 = I$ and $V = I \times e_2$ so $V^2 = I$, and

$$\begin{aligned} [U, V]_+ &= (e_1 \times e_1)(I \times e_2) + (I \times e_2)(e_1 \times e_1) \\ &= e_1 \times e_1 e_2 + e_1 \times e_2 e_1 \\ &= e_1 \times e_1 e_2 - e_1 \times e_1 e_2 \\ &= 0 \end{aligned}$$

So

$$1 \geq < U >^2 + < V >^2 = < e_1 \times e_1 >^2 + < I \times e_2 >^2 = 1 + < I \times e_2 >^2$$

and thus $< I \times e_2 > = 0$.
Similar arguments show that every term in the the expected values $\vec{r} \cdot (I \times \vec{e})$ and $\vec{r'} \cdot (\vec{e}) \times I$ vanishes, and thus so do the expectation values.
To find the probabilities we have the following system of equations.

$$\begin{pmatrix} 1 & & & 1 \\ & 1 & 1 & \\ 1 & 1 & & \\ & & 1 & 1 \\ 1 & & 1 & \\ & 1 & & 1 \end{pmatrix} \begin{pmatrix} p(1,1) \\ p(1,-1) \\ p(-1,1) \\ p(-1,-1) \end{pmatrix} = \begin{pmatrix} (1 - \vec{r} \cdot \vec{r'})/2 \\ (1 + \vec{r} \cdot \vec{r'})/2 \\ 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix}$$

giving

$$\begin{pmatrix} p(1,1) \\ p(1,-1) \\ p(-1,1) \\ p(-1,-1) \end{pmatrix} = \begin{pmatrix} (1 - \vec{r} \cdot \vec{r'})/4 \\ (1 + \vec{r} \cdot \vec{r'})/4 \\ (1 + \vec{r} \cdot \vec{r'})/4 \\ (1 - \vec{r} \cdot \vec{r'})/4 \end{pmatrix}$$

(In the six-row matrix you can omit one row from each complementary pair of rows 3 and 4 and rows 5 and 6.)

d) *Entanglement and EPR.* I have followed [Jor86, Ch.8,Ch.14] for this excursion, making it more concrete and rearranging the argument. In [Jor86, Ch.15,Ch.16] Jordan applies the two-spin results to the famous Einstein-Podolsky-Rosen argument that quantum mechanics is incomplete because it leaves out important aspects of "reality". The EPR argument supposed two particles of opposite spin starting together but then separating and being measured independently after they are too far apart to affect each other, even at lightspeed. Because the pair starts and then continues with spin 0, the particles are "entangled". In both classical and quantum systems we can determine the state of one particle, no matter how far away it is, by measuring the other. In a quantum system we can take a sequence of measurements and the result depends on the sequence, as we saw in part (a) on one spin. With two entangled spins, the sequence of measurements on the particle in Montreal will evidently affect the particle on Betelguese. Read Jordan.

But before you do, here is a summary, which you can now calculate yourself, of the particular directions for which Jordan calculates the probabiliities. The letters $a, b, \alpha, \beta, \gamma, \delta, \epsilon$ labelling the directions are Jordan's: he uses $\{a, \alpha\} \times \{b, \beta\}$ in Chapter 15 on EPR and pairs from $\{\gamma, \delta, \epsilon\}$ in Chapter 16 on John Stuart Bell's inequalities.

$\vec{r}$ $\vec{r'}$	$(1-\vec{r}\cdot\vec{r'})/2$	$p(1,1)$	$p(1,-1)$	$p(-1,1)$	$p(-1,-1)$
a $(0,1,0)$ b $(0,1,0)$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0
a $(0,1,0)$ β $(1,0,0)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
α $(3/5,-4/5,0)$ b $(0,1,0)$	$\frac{9}{10}$	$\frac{9}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{9}{20}$
α $(3/5,-4/5,0)$ β $(1,0,0)$	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{4}{10}$	$\frac{4}{10}$	$\frac{1}{10}$
γ $(0,1,0)$ δ $(\sqrt{3}/2,-1/2,0)$	$\frac{3}{4}$	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{8}$
γ $(0,1,0)$ ϵ $(-\sqrt{3}/2,-1/2,0)$	$\frac{3}{4}$	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{8}$
δ $(\sqrt{3}/2,-1/2,0)$ ϵ $(-\sqrt{3}/2,-1/2,0)$	$\frac{3}{4}$	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{8}$

The upshot of this debate between common-sense causality and quantum mechanics is that quantum mechanics wins. This became clear when Alain Aspect tested the Bell inequalities experimentally against the predictions of quantum mechanics in 1982.

41. **Creation and Annihilation.** We can get two interesting combinations from the nondiagonal Hermitian Pauli matrices (I've changed the sign on one).

$$e = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{and} \qquad f = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$$

Here are the "creation" operator c and the "annihilation" operator a.

$$\begin{aligned} c &= \frac{1}{2}(e+if) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \\ a &= \frac{1}{2}(e-if) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \end{aligned}$$

(Notice that a is just the transpose, or, more generally, the Hermitian conjugate, of c.)
Note that $c^2 = 0 = a^2$ and that

$$ca = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$[c, a]_+ = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Remember (from Excursion *Spin Statistics, etc.* in Week 7a) that there cannot be more than one electron in a state (Pauli's exclusion principle: we could be talking of a state consisting of a given position, momentum and spin). So there are two possible occupancies for a state: 0 or 1.
We're going to re-interpret the 2-dimensional space as representing these two possibilities. (We've also re-interpreted the word "state" as something more physical than the mathematical state, above, in which a matrix is or is not diagonal.)
If $(1,0)^T$ represents "unoccupied" and $(0,1)^T$ represents "occupied" then the effect of c is to put an electron into this state and the effect of a is to take it out. Here are the operations and a new notation.

$c \mid 0 >=\mid 1 >$	$a \mid 1 >=\mid 0 >$
$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

The notation is descriptive: $\mid 0 >$ means unoccupied and $\mid 1 >$ means occupied.
Now let's move on to two states. They could represent different energies, They are (note that the labels are just the binary numbers inverted)

$\mid 0,0 >$	$\mid 1,0 >$	$\mid 0,1 >$	$\mid 1,1 >$
$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

We must find matrices to connect them. There will be four, a creator and an annihilator for each state.

$$\begin{array}{ll} c_0 \mid 0, n_1 >=\mid 1, n_1 > & c_0 \mid 1, n_1 >= 0 \\ a_0 \mid 0, n_1 >= 0 & a_0 \mid 1, n_1 >=\mid 0, n_1 > \\ c_1 \mid n_0, 0 >=\mid n_0, 1 > & c_1 \mid n_1, 1 >= 0 \\ a_1 \mid n_0, 0 >= 0 & a_1 \mid n_0, 1 >=\mid n_0, 0 > \end{array}$$

Here, n_0 and n_1 can be either 0 or 1. And I've been a little hasty: there may need to be $-$ signs after some of the $=$, as we shall see.
Let's start with both states occupied, $\mid 1,1 >$, and swap the two electrons.

$$\mid 1,1 > - - a_1 - - > \mid 1,0 > - - a_0 - - > \mid 0,0 > - - c_1 - - > \mid 0,1 > - - c_0 - - > \mid 1,1 >$$

But because we've swapped the electrons a $-$ sign must appear somewhere in this chain. We'll put it in $c_0 \mid 0,1 > = - \mid 1,1 >$ (but retain $c_0 \mid 0,0 > =\mid 1,0 >$ in the above $c_0 \mid 0, n_1 > = \mid 1, n_1 >$). Correspondingly, because $a = c^\dagger$, we have $a_0 \mid 1,1 > = - \mid 0,1 >$ and keep the other of the pair $a_0 \mid 1, n_1 > = \mid 0, n_1 >$.
The matrices are then $c_0(2,1) = 1, c_0(4,3) = -1, c_1(3,1) = 1$ and $c_1(4,2) = 1$, with the corresponding transposes for a_0 and a_1.
Show that $c_0 a_0 + c_1 a_1$ is the "number operator" and counts the number of occupancies, 0 to

2—it's actually the number of bits set to 1 in the corresponding state label.
Show that $[c_j, c_k]_+ = 0 = [a_j, a_k]_+$ and $[c_j, a_k]_+ = I\delta_{jk}$ where $\delta_{jk} = 0$ unless $j = k$ when it is 1.
Show that $e_j = c_j + a_j$ and $f_j = -i(c_j - a_j)$ form a Clifford algebra, with $e_j^2 = I = f_j^2$ and all e_j and f_k anticommute.
Thus creation and annihilation operators give a way to generate explicit matrix representations for Clifford algebra: this elaborates on Excursion *Matrix representations of Clifford "numbers"* above.
Use the same process to generate 8-by-8 creation and annihilation matrices satisfying the same anticommutation rules, and the corresponding 8-by-8 Clifford algebra. You'll need to put in three minus signs in the c_j: put one in each of the three and make sure none of the index pairs involved are common to any two of the -1 entries, e.g., $c_0(3,2), c_1(6,4), c_2(5,1)$.
[Por04] gives the construction I've followed for creation and annihilation operators and [Dir66, p.26] gives their anticommutator properties.

42. Look up Josiah Willard Gibbs, 1839–1903, and his vector analysis.

43. Look up Sir William Rowan Hamilton, 1805–1865, and his "quaternions". What mental block stumped him for a long time? How did he misinterpret what he invented, and how do quaternions relate to 3D interval algebra? (see [Alt92].)

44. How do the Pauli matrices (Week 6) relate to 3D interval algebra?

45. Why is the number of basic elements of d-dimensional interval algebra equal to 2^d?

46. Survey the usage of the phrase "real world" and identify a legitimate usage distinct from a put-down of academics.

47. Any part of the Preliminary Notes that needs working through.

References

[Alt92] Simon L. Altmann. *Icons and Symmetries.* Clarendon Press, Oxford, 1992.

[Cox63] H. S. M. Coxeter. *Regular Polytopes.* The MacMillan Company, Collier-MacMillan Ltd, New York, London, 1963. 2nd ed.

[Dir66] Paul A. M. Dirac. *Lectures on Quantum Field Theory.* Academic Press Inc, New York and London, 1966. Belfer Graduate School of Science, Yeshiva University, N.Y. lectures 1963–64.

[Har03] James B Hartle. *Gravity: An Introduction to Einstein's General Relativity.* Addison Wesley, San Francisco, 2003.

[Jor86] Thomas F Jordan. *Quantum Mechanics in Simple Matrix Form.* JohnWiley & Sons, Inc., New York, 1986.

[McC57] A. J. McConnell. *Applications of Tensor Analysis.* Dover Publictions, Inc., New York, 1957. Originally *Applications of the Absolute Differential Calculus*, 1931.

[Mer84] T. H. Merrett. *Relational Information Systems.* Reston Publishing Co., Reston, VA., 1984.

[Mer99] T. H. Merrett. Relational information systems. (revisions of [Mer84]):
Data structures for secondary storage: http://www.cs.mcgill.ca/~tim/cs420
Database programming: http://www.cs.mcgill.ca/~tim/cs612, 1999.

[Mer07] T. H. Merrett. Aldat: a retrospective on a work in progress. *Information Systems*, 32(4):505–44, March 2007.

[Por04] F C Porter. Physics 195 course notes: Second quantization. URL http://www.cithep.caltech.edu/∼fcp/physics/quantumMechanics/secondQuantization/SecondQuantization.pdf, last accessed 16/7/12, 2004.