Basic Linear Algebra

This notebook has been written in *Mathematica* by

Mark J. McCready
Professor and Chair of Chemical Engineering
University of Notre Dame
Notre Dame IN 46556
USA

Mark.J.McCready.1@nd.edu
http://www.nd.edu/~mjm/

Version: 8/31/98

This notebook is intended as a companion for the first part of Chapter 1 of:

A. Varma and M. Morbidelli (1997) *Mathematical Methods in Chemical Engineering*, Oxford Press**.**

The notebook shows many of the specific manipulations that we consider as basic linear algebra. It also shows many different techniques for using *Mathematica*.

Chapter 1
Matrices, manipulations and applications

We can first start with a definition of a matrix as a rectangular array of numbers. The horizontal entries are rows and the vertical enties are columns. When we refer to elements of a matrix typically indices such as i,j, k will be used. By convention, the first index is the row and the second index is the column.

■ Some definitions and elementary operations in *Mathematica*

We would like to make a matrix that *Mathematica* can understand. Use an opening {{ then close each row with } and then open the next with { .... after the last row you obviously need a closing }. The one that is defined here has the name "a"

```
a = {{1, 2, 3}, {7, 5, 6}, {1, 6, 9}}
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 7 & 5 & 6 \\ 1 & 6 & 9 \end{pmatrix}$$

The matrix *a* is a 3 by 3 square matrix. We could think of non square matrixes and may encounter them in certain numerical problems. However, let us save these until they become essential.

A matrix could correspond to many different physical entities. For example a 3 by 3 could be the elements in a second order tensor which corresponds to the stresses in a fluid at a point and some time. Of course, if this were the case, the matrix would be symmetric for all simple fluids.

How can we get a symmetric matrix from *a* and what is "symmetric".

First define transpose, which is a switching of a[i,j] --> a[j,i]

```
Transpose[a]
```

$$\begin{pmatrix} 1 & 7 & 1 \\ 2 & 5 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

You see that we left the diagonal elements unchanged and "reflected" the off diagonal elements across the diagonal. For example elements a[i,j] , where i is row and j is column were transformed to a[j,i]

```
a[[1, 2]]
```

2

```
a[[2, 1]]
```

7

Now we can easily get the symmetric part of *a* (such that a[i,j]=a[j,i] and (of course) its antisymmetric piece.

$$\texttt{asymm} = \frac{1}{2}\,(\texttt{a} + \texttt{Transpose[a]})$$

$$\begin{pmatrix} 1 & \frac{9}{2} & 2 \\ \frac{9}{2} & 5 & 6 \\ 2 & 6 & 9 \end{pmatrix}$$

```
aanti = 1/2 (a - Transpose[a])
```

$$\begin{pmatrix} 0 & -\frac{5}{2} & 1 \\ \frac{5}{2} & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

We see that the antisymmetric part does not have any diagonal elements because a[i,i] cannot be equal to -a[i,i]. Thus our definition of an anti symmetric matrix is that a[i,j]=-a[j,i].

Symmetric and antisymetric matrixes are useful, for example, in defining the basic elements of fluid motion in a study of "kinematics". It is convenient that

```
asymm + aanti
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 7 & 5 & 6 \\ 1 & 6 & 9 \end{pmatrix}$$

thus we have recovered *a* again.

From these operations we see that addition and subtraction of matrices is done by simple addition and subtraction of corresponding elements.

```
matrixad[r1_, r2_] := za =
  Table[r1[[i, j]] + r2[[i, j]], {i, 1, Length[r1]}, {j, 1, Length[r2]}]
```

```
a + a^2
```

$$\begin{pmatrix} 2 & 6 & 12 \\ 56 & 30 & 42 \\ 2 & 42 & 90 \end{pmatrix}$$

```
matrixad[a, a^2]
```

$$\begin{pmatrix} 2 & 6 & 12 \\ 56 & 30 & 42 \\ 2 & 42 & 90 \end{pmatrix}$$

```
za
```

$$\begin{pmatrix} 2 & 6 & 12 \\ 56 & 30 & 42 \\ 2 & 42 & 90 \end{pmatrix}$$

To do addition or subtraction, of course, the matrices need to be the same shape.

```
az = {{1, 4}, {4, 5}, {2, 9}}
```

$$\begin{pmatrix} 1 & 4 \\ 4 & 5 \\ 2 & 9 \end{pmatrix}$$

```
a + {{1, 4}, {4, 5}, {2, 9}}
```

— *Thread::tdlen : Objects of unequal length in {1, 4} + {1, 2, 3} cannot be combined.*

— *Thread::tdlen : Objects of unequal length in {4, 5} + {7, 5, 6} cannot be combined.*

— *Thread::tdlen : Objects of unequal length in {2, 9} + {1, 6, 9} cannot be combined.*

— *General::stop : Further output of Thread::tdlen will be suppressed during this calculation.*

{{1, 4} + {1, 2, 3}, {4, 5} + {7, 5, 6}, {2, 9} + {1, 6, 9}}

Even *Mathematica* recognizes the problem.

■ Index notation

In mechanics, many physical quantities are represented by rectangular arrays of numbers. For example, mass or speed are scalars ====> m, s

velocity or acceleration are normally vectors $a = (a_1, a_2, a_3)$

If you wanted to describe the velocity gradients in a fluid, at a point, at some instant in time, you would write the tensor *grad u*, where *grad* and *u* are both vectors, the result (a *dyad* product) is a 3 by 3 array which is a second order tensor.

Other entities lead to higher order tensors.

In all of these, there are only three space directions. Thus we can use "1, 2, 3" as the indices, instead of say, x, y, z. If this is done we can think of representing scalars, vectors and tensors with the same notation as

scalar ===> s, zero order tensor

vector ===> v_i , first order tensor

"tensor"===> T_{ij}, second order tensor

third order tensor , Q_{ijk},

etc.

We use the convention that each subscript takes on all values of 1, 2, 3, $v_i = (v_1, v_2, v_3)$

For vector operations, we use the convention that repeated indexes are summed over

$v_i u_i = v_1 u_1 + v_2 u_2 + v_3 u_3 = \mathbf{v} \cdot \mathbf{u}$

■ How do we do a calculation, and how do we do it in *Mathematica?*

■ Cross Product

Let's consider a cross product in index notation. We have the definition, $\epsilon_{\text{ijk}} u_j v_k$.

Here is how we do this calculation

```
cross1 =
 Table[Sum[Sum[ϵ[i, j, k] u[j] v[k], {k, 1, 3}], {j, 1, 3}], {i, 1, 3}]
```

$$\{u(1)\,v(1)\,\epsilon(1,\,1,\,1) + u(1)\,v(2)\,\epsilon(1,\,1,\,2) + u(1)\,v(3)\,\epsilon(1,\,1,\,3) + u(2)\,v(1)\,\epsilon(1,\,2,\,1) + u(2)\,v(2)\,\epsilon(1,\,2,\,2) + u(2)\,v(3)\,\epsilon(1,\,2,\,3) + u(3)\,v(1)\,\epsilon(1,\,3,\,1) + u(3)\,v(2)\,\epsilon(1,\,3,\,2) + u(3)\,v(3)\,\epsilon(1,\,3,\,3),\ u(1)\,v(1)\,\epsilon(2,\,1,\,1) + u(1)\,v(2)\,\epsilon(2,\,1,\,2) + u(1)\,v(3)\,\epsilon(2,\,1,\,3) + u(2)\,v(1)\,\epsilon(2,\,2,\,1) + u(2)\,v(2)\,\epsilon(2,\,2,\,2) + u(2)\,v(3)\,\epsilon(2,\,2,\,3) + u(3)\,v(1)\,\epsilon(2,\,3,\,1) + u(3)\,v(2)\,\epsilon(2,\,3,\,2) + u(3)\,v(3)\,\epsilon(2,\,3,\,3),\ u(1)\,v(1)\,\epsilon(3,\,1,\,1) + u(1)\,v(2)\,\epsilon(3,\,1,\,2) + u(1)\,v(3)\,\epsilon(3,\,1,\,3) + u(2)\,v(1)\,\epsilon(3,\,2,\,1) + u(2)\,v(2)\,\epsilon(3,\,2,\,2) + u(2)\,v(3)\,\epsilon(3,\,2,\,3) + u(3)\,v(1)\,\epsilon(3,\,3,\,1) + u(3)\,v(2)\,\epsilon(3,\,3,\,2) + u(3)\,v(3)\,\epsilon(3,\,3,\,3)\}$$

We know that we sum over repreated indices and let single ones take any value. Sum[#, {j,1,3}] takes care of the repeated index j. A Table[#,{i,1,3}] takes care of a free index.

Let's do the calculation one step at a time. I think that it is easiest to start with the "rightmost" repeated index. This is k. We first sum over k.

```
tempk = Sum[ϵ[i, j, k] u[j] v[k], {k, 1, 3}]
```

$$u(j)\,v(1)\,\epsilon(i,\,j,\,1) + u(j)\,v(2)\,\epsilon(i,\,j,\,2) + u(j)\,v(3)\,\epsilon(i,\,j,\,3)$$

We see how the sum works. You leave i and j alone and then let k take values of 1,2,3 -- and sum these. There are now 3 terms. You still have 2 free indices.

Now we need to sum over j

```
tempj = Sum[tempk, {j, 1, 3}]
```

— *General::spell1 : Possible spelling error: new symbol name "tempj" is similar to existing symbol "tempk".*

$$u(1)\,v(1)\,\epsilon(i,\,1,\,1) + u(1)\,v(2)\,\epsilon(i,\,1,\,2) + u(1)\,v(3)\,\epsilon(i,\,1,\,3) + u(2)\,v(1)\,\epsilon(i,\,2,\,1) + u(2)\,v(2)\,\epsilon(i,\,2,\,2) + u(2)\,v(3)\,\epsilon(i,\,2,\,3) + u(3)\,v(1)\,\epsilon(i,\,3,\,1) + u(3)\,v(2)\,\epsilon(i,\,3,\,2) + u(3)\,v(3)\,\epsilon(i,\,3,\,3)$$

We have created a single sum with 9 terms now. If you summed over i, (which is wrong) you would have a scalar with 27 terms. What we do is use the Table command to create a vector with three elements. Note the commas that separate the elements.

```
tempi = Table[tempj, {i, 1, 3}]
```

— *General::spell : Possible spelling error: new symbol name "tempi" is similar to existing symbols {tempj, tempk}.*

$\{u(1)\, v(1)\, \epsilon(1, 1, 1) + u(1)\, v(2)\, \epsilon(1, 1, 2) + u(1)\, v(3)\, \epsilon(1, 1, 3) + u(2)\, v(1)\, \epsilon(1, 2, 1) + u(2)\, v(2)\, \epsilon(1, 2, 2) + u(2)\, v(3)\, \epsilon(1, 2, 3) + u(3)\, v(1)\, \epsilon(1, 3, 1) + u(3)\, v(2)\, \epsilon(1, 3, 2) + u(3)\, v(3)\, \epsilon(1, 3, 3),\ u(1)\, v(1)\, \epsilon(2, 1, 1) + u(1)\, v(2)\, \epsilon(2, 1, 2) + u(1)\, v(3)\, \epsilon(2, 1, 3) + u(2)\, v(1)\, \epsilon(2, 2, 1) + u(2)\, v(2)\, \epsilon(2, 2, 2) + u(2)\, v(3)\, \epsilon(2, 2, 3) + u(3)\, v(1)\, \epsilon(2, 3, 1) + u(3)\, v(2)\, \epsilon(2, 3, 2) + u(3)\, v(3)\, \epsilon(2, 3, 3),\ u(1)\, v(1)\, \epsilon(3, 1, 1) + u(1)\, v(2)\, \epsilon(3, 1, 2) + u(1)\, v(3)\, \epsilon(3, 1, 3) + u(2)\, v(1)\, \epsilon(3, 2, 1) + u(2)\, v(2)\, \epsilon(3, 2, 2) + u(2)\, v(3)\, \epsilon(3, 2, 3) + u(3)\, v(1)\, \epsilon(3, 3, 1) + u(3)\, v(2)\, \epsilon(3, 3, 2) + u(3)\, v(3)\, \epsilon(3, 3, 3)\}$

We know the values of the ϵ[i,j,k]'s. Let's first define all of these (using brute force) and then use a substitution command to apply them to "tempi".

Here we can define the alternating unit tensor, eps[i,j,k]. (Note that the semicolon suppresses the output.

```
eps[1, 2, 3] = 1;
eps[2, 3, 1] = 1;
eps[3, 1, 2] = 1;
eps[3, 2, 1] = -1;
eps[2, 1, 3] = -1;
eps[1, 3, 2] = -1;
eps[1, 1, 1] = 0;
eps[2, 2, 2] = 0;
eps[3, 3, 3] = 0;
eps[1, 1, 2] = 0;
eps[1, 1, 3] = 0;
eps[2, 1, 2] = 0;
eps[2, 1, 1] = 0;
eps[2, 2, 1] = 0;
eps[1, 2, 2] = 0;
eps[1, 3, 3] = 0;
eps[3, 1, 1] = 0;
eps[3, 3, 1] = 0;
eps[3, 3, 2] = 0;
eps[3, 2, 2] = 0;
eps[2, 3, 3] = 0;
eps[1, 2, 1] = 0;
eps[1, 3, 1] = 0;
eps[2, 3, 2] = 0;
eps[3, 2, 3] = 0;
eps[3, 1, 3] = 0;
eps[2, 2, 3] = 0;
```

To do the substitution, we make a list with one level of braces "{". It takes a couple of Flatten commands which each reduce the number of braces around an object.

```
sublist = Flatten[Flatten[
   Table[ϵ[i, j, k] -> eps[i, j, k], {i, 1, 3}, {j, 1, 3}, {k, 1, 3}]]]
```

$\{\epsilon(1, 1, 1) \to 0, \epsilon(1, 1, 2) \to 0, \epsilon(1, 1, 3) \to 0, \epsilon(1, 2, 1) \to 0, \epsilon(1, 2, 2) \to 0,$
$\epsilon(1, 2, 3) \to 1, \epsilon(1, 3, 1) \to 0, \epsilon(1, 3, 2) \to -1, \epsilon(1, 3, 3) \to 0, \epsilon(2, 1, 1) \to 0,$
$\epsilon(2, 1, 2) \to 0, \epsilon(2, 1, 3) \to -1, \epsilon(2, 2, 1) \to 0, \epsilon(2, 2, 2) \to 0, \epsilon(2, 2, 3) \to 0,$
$\epsilon(2, 3, 1) \to 1, \epsilon(2, 3, 2) \to 0, \epsilon(2, 3, 3) \to 0, \epsilon(3, 1, 1) \to 0, \epsilon(3, 1, 2) \to 1, \epsilon(3, 1, 3) \to 0,$
$\epsilon(3, 2, 1) \to -1, \epsilon(3, 2, 2) \to 0, \epsilon(3, 2, 3) \to 0, \epsilon(3, 3, 1) \to 0, \epsilon(3, 3, 2) \to 0, \epsilon(3, 3, 3) \to 0\}$

Now we use it on tempi with the "/." (this substitution coomand must be a secret code from Steve Wolfram's first computer algebra package.

```
crossproduct = tempi /. sublist
```

$\{u(2)\, v(3) - u(3)\, v(2), u(3)\, v(1) - u(1)\, v(3), u(1)\, v(2) - u(2)\, v(1)\}$

We can check this against the canned function,

```
check = Cross[{u[1], u[2], u[3]}, {v[1], v[2], v[3]}]
```

— *General::spell1 : Possible spelling error: new symbol name "check" is similar to existing symbol "Check".*

$\{u(2)\, v(3) - u(3)\, v(2), u(3)\, v(1) - u(1)\, v(3), u(1)\, v(2) - u(2)\, v(1)\}$

It works!!

```
crossproduct - check
```

{0, 0, 0}

We might also look at the definition of the Cross Product using the Determinant.

```
det1 = {{e[1], e[2], e[3]}, {u[1], u[2], u[3]}, {v[1], v[2], v[3]}}
```

$$\begin{pmatrix} e(1) & e(2) & e(3) \\ u(1) & u(2) & u(3) \\ v(1) & v(2) & v(3) \end{pmatrix}$$

```
det2 = Det[det1]
```

$-e(3)\, u(2)\, v(1) + e(2)\, u(3)\, v(1) + e(3)\, u(1)\, v(2) - e(1)\, u(3)\, v(2) - e(2)\, u(1)\, v(3) + e(1)\, u(2)\, v(3)$

We get this in the normal form by finding the coeffients of the terms multiplying each of the e[i] unit vectors,

```
det3 = Table[Coefficient[det2, e[i]], {i, 1, 3}]
```

$\{u(2)\, v(3) - u(3)\, v(2), u(3)\, v(1) - u(1)\, v(3), u(1)\, v(2) - u(2)\, v(1)\}$

```
det3 - check
```

{0, 0, 0}

- **Dot product of two vectors, $v_i\ u_i$.**

Just sum over i.

```
Sum[v[i] u[i], {i, 1, 3}]
```

$u(1)\,v(1) + u(2)\,v(2) + u(3)\,v(3)$

```
Dot[{v[1], v[2], v[3]}, {u[1], u[2], u[3]}]
```

$u(1)\,v(1) + u(2)\,v(2) + u(3)\,v(3)$

- **Here is a dot product of a vector with a tensor, $u_i\ T_{ij}$**

Sum over i and let j take all three values,

```
Table[Sum[u[i] T[i, j], {i, 1, 3}], {j, 1, 3}]
```

$\{T(1,1)\,u(1) + T(2,1)\,u(2) + T(3,1)\,u(3),\ T(1,2)\,u(1) + T(2,2)\,u(2) + T(3,2)\,u(3),\ T(1,3)\,u(1) + T(2,3)\,u(2) + T(3,3)\,u(3)\}$

We can make this look better as a column vector using the command,

```
MatrixForm[%]
```

$$\begin{pmatrix} T(1,1)\,u(1) + T(2,1)\,u(2) + T(3,1)\,u(3) \\ T(1,2)\,u(1) + T(2,2)\,u(2) + T(3,2)\,u(3) \\ T(1,3)\,u(1) + T(2,3)\,u(2) + T(3,3)\,u(3) \end{pmatrix}$$

Again it might be useful to see the intermediate step, so we redo it:

```
q1 = Sum[u[i] T[i, j], {i, 1, 3}]
```

$T(1,\ j)\,u(1) + T(2,\ j)\,u(2) + T(3,\ j)\,u(3)$

Then do the Table to get the answer

```
q2 = Table[q1, {j, 1, 3}]
```

$\{T(1,1)\,u(1) + T(2,1)\,u(2) + T(3,1)\,u(3),\ T(1,2)\,u(1) + T(2,2)\,u(2) + T(3,2)\,u(3),\ T(1,3)\,u(1) + T(2,3)\,u(2) + T(3,3)\,u(3)\}$

- **Here is the "double dot" product of two tensors, $T_{ij}\ S_{ji}$ = T : S**

We just need a double sum

```
Sum[Sum[T[i, j] S[j, i], {i, 1, 3}], {j, 1, 3}]
```

$S(1,1)\,T(1,1) + S(2,1)\,T(1,2) + S(3,1)\,T(1,3) + S(1,2)\,T(2,1) + S(2,2)\,T(2,2) + S(3,2)\,T(2,3) + S(1,3)\,T(3,1) + S(2,3)\,T(3,2) + S(3,3)\,T(3,3)$

```
temps = Sum[ T[i, j] S[j, i], {i, 1, 3}]
```

— *General::spell : Possible spelling error: new symbol name "temps" is similar to existing symbols {tempi, tempj, tempk}.*

$S(j, 1)\, T(1, j) + S(j, 2)\, T(2, j) + S(j, 3)\, T(3, j)$

```
Sum[temps, {j, 1, 3}]
```

$S(1, 1)\, T(1, 1) + S(2, 1)\, T(1, 2) + S(3, 1)\, T(1, 3) + S(1, 2)\, T(2, 1) + S(2, 2)\, T(2, 2) + S(3, 2)\, T(2, 3) + S(1, 3)\, T(3, 1) + S(2, 3)\, T(3, 2) + S(3, 3)\, T(3, 3)$

■ Multiplication

Any matrix can be multiplied by a single scalar constant.

Define a new matrix rr, note that the semicolon suppresses display

```
rr = {{r11, r12, r13}, {r21, r22, r23}, {r31, r32, r33}};
```

To multiply we have equivalently:

```
α rr
```

$$\begin{pmatrix} \alpha r_{11} & \alpha r_{12} & \alpha r_{13} \\ \alpha r_{21} & \alpha r_{22} & \alpha r_{23} \\ \alpha r_{31} & \alpha r_{32} & \alpha r_{33} \end{pmatrix}$$

```
rr α
```

$$\begin{pmatrix} \alpha r_{11} & \alpha r_{12} & \alpha r_{13} \\ \alpha r_{21} & \alpha r_{22} & \alpha r_{23} \\ \alpha r_{31} & \alpha r_{32} & \alpha r_{33} \end{pmatrix}$$

If we want to multiply matries, we need to be careful of the definition.

Consider a degenerate matrix, a vector, that is also a first order tensor (assuming a physical orgin)

```
ss = {s1, s2, s3};
tt = {t1, t2, t3};
```

A dot product yields: (s_i t_i)

```
ss . tt
```

$s_1\, t_1 + s_2\, t_2 + s_3\, t_3$

which is the same as: (t_i s_i)

```
tt . ss
```

$s_1\ t_1 + s_2\ t_2 + s_3\ t_3$

```
Dot[ss, tt]
```

$s_1\ t_1 + s_2\ t_2 + s_3\ t_3$

However if we are tempted to use a * for multiplication, we get a vector back but this is not a normal operation. (note it is not ($s_i\ t_j$) which is a second order tensor.

```
ss * tt
```

$\{s_1\ t_1, s_2\ t_2, s_3\ t_3\}$

This is intepretted the same as ss tt

```
ss tt
```

$\{s_1\ t_1, s_2\ t_2, s_3\ t_3\}$

```
tt ss
```

$\{s_1\ t_1, s_2\ t_2, s_3\ t_3\}$

Please note that Dot is distict from, *, which gives a product that usually does not arise in physical problems.

There is also a Cross Product but we will save this for later.

You may recognize that the "dot" is giving the standard type of multiplication that we define for matrices (rr_{ij} qq_{jk})

```
qq = {{q11, q12, q13}, {q21, q22, q23}, {q31, q32, q33}};
rr . qq
```

$$\begin{pmatrix} q_{11}\ r_{11} + q_{21}\ r_{12} + q_{31}\ r_{13} & q_{12}\ r_{11} + q_{22}\ r_{12} + q_{32}\ r_{13} & q_{13}\ r_{11} + q_{23}\ r_{12} + q_{33}\ r_{13} \\ q_{11}\ r_{21} + q_{21}\ r_{22} + q_{31}\ r_{23} & q_{12}\ r_{21} + q_{22}\ r_{22} + q_{32}\ r_{23} & q_{13}\ r_{21} + q_{23}\ r_{22} + q_{33}\ r_{23} \\ q_{11}\ r_{31} + q_{21}\ r_{32} + q_{31}\ r_{33} & q_{12}\ r_{31} + q_{22}\ r_{32} + q_{32}\ r_{33} & q_{13}\ r_{31} + q_{23}\ r_{32} + q_{33}\ r_{33} \end{pmatrix}$$

This could be quite different from (qq_{ij} rr_{jk})

```
qq . rr
```

$$\begin{pmatrix} q_{11}\ r_{11} + q_{12}\ r_{21} + q_{13}\ r_{31} & q_{11}\ r_{12} + q_{12}\ r_{22} + q_{13}\ r_{32} & q_{11}\ r_{13} + q_{12}\ r_{23} + q_{13}\ r_{33} \\ q_{21}\ r_{11} + q_{22}\ r_{21} + q_{23}\ r_{31} & q_{21}\ r_{12} + q_{22}\ r_{22} + q_{23}\ r_{32} & q_{21}\ r_{13} + q_{22}\ r_{23} + q_{23}\ r_{33} \\ q_{31}\ r_{11} + q_{32}\ r_{21} + q_{33}\ r_{31} & q_{31}\ r_{12} + q_{32}\ r_{22} + q_{33}\ r_{32} & q_{31}\ r_{13} + q_{32}\ r_{23} + q_{33}\ r_{33} \end{pmatrix}$$

Note that if we forget the . then the result is

```
rr qq
```

$$\begin{pmatrix} q_{11}\, r_{11} & q_{12}\, r_{12} & q_{13}\, r_{13} \\ q_{21}\, r_{21} & q_{22}\, r_{22} & q_{23}\, r_{23} \\ q_{31}\, r_{31} & q_{32}\, r_{32} & q_{33}\, r_{33} \end{pmatrix}$$

Note also that matices do not have to be square to be multiplied, but it has to work

```
rr . tt
```

$$\{r_{11}\, t_1 + r_{12}\, t_2 + r_{13}\, t_3,\ r_{21}\, t_1 + r_{22}\, t_2 + r_{23}\, t_3,\ r_{31}\, t_1 + r_{32}\, t_2 + r_{33}\, t_3\}$$

If we interpret t as a column vector, then this should not work. ***Mathematica* does not care!!!**

```
tt . rr
```

$$\{r_{11}\, t_1 + r_{21}\, t_2 + r_{31}\, t_3,\ r_{12}\, t_1 + r_{22}\, t_2 + r_{32}\, t_3,\ r_{13}\, t_1 + r_{23}\, t_2 + r_{33}\, t_3\}$$

The associative and distributive laws also work for matrices, (just not the commutative)

■ Identity matix

In scalar space, we can't comfortably talk about multiplication until we know what "entity" gives leaves a number unchanged in multiplication (i.e., 1). For a matrix we need:

```
ii = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}
```

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

```
rr . ii
```

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

```
ii . rr
```

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

■ determinants and cofactors

Perhaps your favorite operation on an matrix is the determinant, (which incidentally does not change value if a matix has its coordinate system rotated). The determinant is a scalar and can be easily calculated from the matrix elements.

If the "matrix" has only one element then

```
ss = {{ss1}};
Det[ss]
```

ss1

for a 2 by 2 we have:

```
ww = {{w11, w12}, {w21, w22}};
Det[ww]
```

$w_{11}\, w_{22} - w_{12}\, w_{21}$

For a 3 by 3 we have

```
Det[rr]
```

$-r_{13}\, r_{22}\, r_{31} + r_{12}\, r_{23}\, r_{31} + r_{13}\, r_{21}\, r_{32} - r_{11}\, r_{23}\, r_{32} - r_{12}\, r_{21}\, r_{33} + r_{11}\, r_{22}\, r_{33}$

This looks like a mess but there is a system

To get a general definition of the determinant, it is convenient to define minors and cofactors.

We do this following Varma and Morbidelli. *Mathematica* has a different idea of a cofactor. Thus I need to make my own definition.

det[a_] := Sum[a[[1,i]] cofactor[a] [[1,i]],{i,1,Length[a]}]

minors = m[[i,j]] are the determinants that remain when the row and column that contains an element is crossed out.
cofactors are (-1)^(i+j) m[[i,j]]

You can see how we can define a determinant of arbitrary size using this definition.

Let's try some definitions

First let us see what *Mathematica* does for us, Recall ww

```
ww
```

$$\begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}$$

```
Minors[ww, 1]
```

$$\begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}$$

As I mention above this is not the normal definition of Minors. We want the answer with the element that is crossed out, not vice versa. Thus we can define our own function to *flip* the elements. This is not hard but looks complicated.

```
flip[a_] := Table[zzx[i, j] = a⟦Length[a] - i + 1, Length[a] - j + 1⟧,
  {i, 1, Length[a]}, {j, 1, Length[a]}]
```

We want and get the correct minors from

```
flip[Minors[ww, 1]]
```

$$\begin{pmatrix} w_{22} & w_{21} \\ w_{12} & w_{11} \end{pmatrix}$$

```
flip[Minors[rr, 2]]
```

$$\begin{pmatrix} r_{22} r_{33} - r_{23} r_{32} & r_{21} r_{33} - r_{23} r_{31} & r_{21} r_{32} - r_{22} r_{31} \\ r_{12} r_{33} - r_{13} r_{32} & r_{11} r_{33} - r_{13} r_{31} & r_{11} r_{32} - r_{12} r_{31} \\ r_{12} r_{23} - r_{13} r_{22} & r_{11} r_{23} - r_{13} r_{21} & r_{11} r_{22} - r_{12} r_{21} \end{pmatrix}$$

If we want the cofactors we could define

```
cofactor[a_] := Table[flip[Minors[a, Length[a] - 1]]⟦i, j⟧ (-1)^(i+j),
  {i, 1, Length[a]}, {j, 1, Length[a]}]
```

```
cofactor[ww]
```

$$\begin{pmatrix} w_{22} & -w_{21} \\ -w_{12} & w_{11} \end{pmatrix}$$

```
cofactor[rr]
```

$$\begin{pmatrix} r_{22} r_{33} - r_{23} r_{32} & r_{23} r_{31} - r_{21} r_{33} & r_{21} r_{32} - r_{22} r_{31} \\ r_{13} r_{32} - r_{12} r_{33} & r_{11} r_{33} - r_{13} r_{31} & r_{12} r_{31} - r_{11} r_{32} \\ r_{12} r_{23} - r_{13} r_{22} & r_{13} r_{21} - r_{11} r_{23} & r_{11} r_{22} - r_{12} r_{21} \end{pmatrix}$$

This looks OK, let us check it with the determinant calculated by the LaPlace expansion (p. 6 in V&M)

```
det[a_] := Sum[a[[1, i]] cofactor[a][[1, i]], {i, 1, Length[a]}]
```

$$\text{det}[a\_] := \sum_{i=1}^{\text{Length}[a]} a[\![1, i]\!]\ \text{cofactor}[a][\![1, i]\!]$$

— *General::spell1 : Possible spelling error: new symbol name "det" is similar to existing symbol "Det".*

```
det[rr]
```

$r_{13} (r_{21} r_{32} - r_{22} r_{31}) + r_{12} (r_{23} r_{31} - r_{21} r_{33}) + r_{11} (r_{22} r_{33} - r_{23} r_{32})$

$$\sum_{i=1}^{\text{Length}[a]} rr[\![2, i]\!]\ \text{cofactor}[rr][\![2, i]\!]$$

$r_{23} (r_{12} r_{31} - r_{11} r_{32}) + r_{22} (r_{11} r_{33} - r_{13} r_{31}) + r_{21} (r_{13} r_{32} - r_{12} r_{33})$

```
Simplify[% - %29]
```

0

We can check it against the real function

```
Expand[det[rr] - Det[rr]]
```

0

Perhaps we should check it with a bigger Matrix

```
z4 = {{1, 3, 5, 7}, {6, 6, 5, 4}, {1, 1, 26, 4}, {1, 8, 0, 3}}
```

$$\begin{pmatrix} 1 & 3 & 5 & 7 \\ 6 & 6 & 5 & 4 \\ 1 & 1 & 26 & 4 \\ 1 & 8 & 0 & 3 \end{pmatrix}$$

```
Det[z4]
```

5373

```
det[z4]
```

5373

OK, now we see how this can give the determinant of an arbitrary size square matrix.

We also note that the formula for calculating the determinant is known as the LaPlace expansion. We could calculate the determinant from any row of the matrix because:

```
Det[a]
```

6

Use first row:

```
Length[a]
  ∑      a〚1, i〛 cofactor[a]〚1, i〛
 i=1
```

6

Use second row:

```
Length[a]
  ∑      a〚2, i〛 cofactor[a]〚2, i〛
 i=1
```

6

Use third row:

```
Length[a]
  ∑      a〚3, i〛 cofactor[a]〚3, i〛
 i=1
```

6

The answer is the same.

■ Rank

The rank of a matrix is the size of the largest non-zero determinant that can be obtained from it. The rank has important implications when one wants to solve systems of linear equations.

Specifically, see page 16 of V&M.

■ Systems of equations

A big application area of matices for us will be systems of linear equations. Suppose we have the system:
$R\,x == y$, where R is a square matix and x and y are vectors. We could write;

$R\,x - y == 0$

Define

```
xx = {x1, x2, x3};
yy = {y1, y2, y3};
```

In our convenient shorthand we have

```
rr . xx - yy
```

$\{r_{11}\,x_1 + r_{12}\,x_2 + r_{13}\,x_3 - y_1,\ r_{21}\,x_1 + r_{22}\,x_2 + r_{23}\,x_3 - y_2,\ r_{31}\,x_1 + r_{32}\,x_2 + r_{33}\,x_3 - y_3\}$

Which is a list of three equations that equal 0.

■ Inverse matrix

Now that we have dared to write the matrix form of a system of equations, we cannot rest until we have a solution.

For *R x* == *y*, we want Inverse[R] also written R^{-1}, that allows
us to get *x* from
Inverse[*R*] *R x* = Inverse[*R*] *y* =*x*

```
Inverse[rr] . rr
```

$$\begin{pmatrix} \frac{(r_{12}\, r_{23}-r_{13}\, r_{22})\, r_{31}}{-r_{13}\, r_{22}\, r_{31}+r_{12}\, r_{23}\, r_{31}+r_{13}\, r_{21}\, r_{32}-r_{11}\, r_{23}\, r_{32}-r_{12}\, r_{21}\, r_{33}+r_{11}\, r_{22}\, r_{33}} + \frac{r_{21}\, (r_{13}\, r_{32}-r_{12}\, r_{33})}{-r_{13}\, r_{22}\, r_{31}+r_{12}\, r_{23}\, r_{31}+r_{13}\, r_{21}\, r_{32}-r_{11}\, r_{23}\, r_{32}-r_{12}\, r_{21}\, r_{33}+r_{11}\, r_{22}\, r_{33}} + \frac{}{-r_{13} \cdots} \\ \frac{(r_{13}\, r_{21}-r_{11}\, r_{23})\, r_{31}}{-r_{13}\, r_{22}\, r_{31}+r_{12}\, r_{23}\, r_{31}+r_{13}\, r_{21}\, r_{32}-r_{11}\, r_{23}\, r_{32}-r_{12}\, r_{21}\, r_{33}+r_{11}\, r_{22}\, r_{33}} + \frac{r_{21}\, (r_{11}\, r_{33}-r_{13}\, r_{31})}{-r_{13}\, r_{22}\, r_{31}+r_{12}\, r_{23}\, r_{31}+r_{13}\, r_{21}\, r_{32}-r_{11}\, r_{23}\, r_{32}-r_{12}\, r_{21}\, r_{33}+r_{11}\, r_{22}\, r_{33}} + \frac{}{-r_{13} \cdots} \\ \frac{(r_{11}\, r_{22}-r_{12}\, r_{21})\, r_{31}}{-r_{13}\, r_{22}\, r_{31}+r_{12}\, r_{23}\, r_{31}+r_{13}\, r_{21}\, r_{32}-r_{11}\, r_{23}\, r_{32}-r_{12}\, r_{21}\, r_{33}+r_{11}\, r_{22}\, r_{33}} + \frac{r_{21}\, (r_{12}\, r_{31}-r_{11}\, r_{32})}{-r_{13}\, r_{22}\, r_{31}+r_{12}\, r_{23}\, r_{31}+r_{13}\, r_{21}\, r_{32}-r_{11}\, r_{23}\, r_{32}-r_{12}\, r_{21}\, r_{33}+r_{11}\, r_{22}\, r_{33}} + \frac{}{-r_{13} \cdots} \end{pmatrix}$$

```
Simplify[%]
```

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

Which is the identity matix and thus the solution vector for x is

```
FullSimplify[Inverse[rr] . yy]
```

$$\begin{pmatrix} \frac{r_{22}\, r_{33}-r_{23}\, r_{32}}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+r_{12}\, (r_{23}\, r_{31}-r_{21}\, r_{33})+r_{11}\, (r_{22}\, r_{33}-r_{23}\, r_{32})} & \frac{r_{13}\, r_{32}-r_{12}\, r_{33}}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+r_{12}\, (r_{23}\, r_{31}-r_{21}\, r_{33})+r_{11}\, (r_{22}\, r_{33}-r_{23}\, r_{32})} & \frac{}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+\cdots} \\ \frac{r_{23}\, r_{31}-r_{21}\, r_{33}}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+r_{12}\, (r_{23}\, r_{31}-r_{21}\, r_{33})+r_{11}\, (r_{22}\, r_{33}-r_{23}\, r_{32})} & \frac{r_{11}\, r_{33}-r_{13}\, r_{31}}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+r_{12}\, (r_{23}\, r_{31}-r_{21}\, r_{33})+r_{11}\, (r_{22}\, r_{33}-r_{23}\, r_{32})} & \frac{}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+\cdots} \\ \frac{r_{21}\, r_{32}-r_{22}\, r_{31}}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+r_{12}\, (r_{23}\, r_{31}-r_{21}\, r_{33})+r_{11}\, (r_{22}\, r_{33}-r_{23}\, r_{32})} & \frac{r_{12}\, r_{31}-r_{11}\, r_{32}}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+r_{12}\, (r_{23}\, r_{31}-r_{21}\, r_{33})+r_{11}\, (r_{22}\, r_{33}-r_{23}\, r_{32})} & \frac{}{r_{13}\, (r_{21}\, r_{32}-r_{22}\, r_{31})+\cdots} \end{pmatrix} . \, yy$$

Perhaps this is cleaner numerically, a x == z

```
z = {1, 5, 9};
```

```
Inverse[a]
```

$$\begin{pmatrix} \frac{3}{2} & 0 & -\frac{1}{2} \\ -\frac{19}{2} & 1 & \frac{5}{2} \\ \frac{37}{6} & -\frac{2}{3} & -\frac{3}{2} \end{pmatrix}$$

```
Inverse[a] . a
```

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

```
Inverse[a] . z
```

$$\left\{-3, 18, -\frac{32}{3}\right\}$$

Which can be obtained equivalently from

```
LinearSolve[a, z]
```

$$\left\{-3, 18, -\frac{32}{3}\right\}$$

```
Information["LinearSolve", LongForm → True]
```

LinearSolve[m, b] finds an x which solves the matrix equation m.x==b.

Attributes[LinearSolve] = {Protected}

Options[LinearSolve] = {Method -> Automatic, Modulus -> 0, ZeroTest -> (#1 == 0 &)}

Of course, *a* better have an inverse for this to work!!

How will we calculate the inverse matrix? Set up the problem in terms of the result that you need. (A famous trick from mathematics!!)

Here is our unknown matrix

```
wwinv = {{winv11, winv12}, {winv21, winv22}}
```

$$\begin{pmatrix} \text{winv}_{11} & \text{winv}_{12} \\ \text{winv}_{21} & \text{winv}_{22} \end{pmatrix}$$

We can construct the matrix that will have to give the indentity matrix.

```
eqs = wwinv . ww
```

$$\begin{pmatrix} w_{11}\,\text{winv}_{11} + w_{21}\,\text{winv}_{12} & w_{12}\,\text{winv}_{11} + w_{22}\,\text{winv}_{12} \\ w_{11}\,\text{winv}_{21} + w_{21}\,\text{winv}_{22} & w_{12}\,\text{winv}_{21} + w_{22}\,\text{winv}_{22} \end{pmatrix}$$

This is easily solved

```
Solve[eqs == IdentityMatrix[2], {winv11, winv12, winv21, winv22}]
```

$$\left\{\left\{\text{winv}_{11} \to \frac{w_{22}}{w_{11}\, w_{22} - w_{12}\, w_{21}}, \text{winv}_{12} \to -\frac{w_{12}}{w_{11}\, w_{22} - w_{12}\, w_{21}}, \text{winv}_{21} \to -\frac{w_{21}}{w_{11}\, w_{22} - w_{12}\, w_{21}}, \text{winv}_{22} \to \frac{w_{11}}{w_{11}\, w_{22} - w_{12}\, w_{21}}\right\}\right\}$$

This matches the built in function.

```
Inverse[ww]
```

$$\begin{pmatrix} \frac{w_{22}}{w_{11}\, w_{22}-w_{12}\, w_{21}} & -\frac{w_{12}}{w_{11}\, w_{22}-w_{12}\, w_{21}} \\ -\frac{w_{21}}{w_{11}\, w_{22}-w_{12}\, w_{21}} & \frac{w_{11}}{w_{11}\, w_{22}-w_{12}\, w_{21}} \end{pmatrix}$$

Here is a defined function that calculates the inverse from cofactors and the determinant

```
inverse[a_] := Transpose[cofactor[a]] / det[a]
```

```
Simplify[inverse[ww] - Inverse[ww]]
```

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

```
Simplify[inverse[a] - Inverse[a]]
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
Simplify[inverse[z4] - Inverse[z4]]
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Thus the matrix that defines the inverse is the Transpose[Matrix of cofactors]/Determinant!!

■ Important point

The term "adjoint" has dual meanings. Varma and Morbidelli and Noble and Daniel, for example, call the Transpose[cofactor[a]] the "adjoint" matrix. In the context of linear operator theory, the adjoint for a constant matrix is adjoint[a] = Conjugate[Transpose[a]]. These two uses are not the same. We will make every effort to not confuse these in class, homework and tests.

■ Gaussian elimination

From a practical point, how is an inverse found or how do we solve a system of equations? The standard procedure is *Gaussian Elimination*. You probably have used it before so only a quick example will be given.

A good reference for the next few sections is:
B. Noble and J. W. Daniel, (1988) *Applied Linear Algebra*, Prentice-Hall.

Consider A x = b

```
a = {{1, 2, 3}, {7, 5, 6}, {1, 6, 9}}
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 7 & 5 & 6 \\ 1 & 6 & 9 \end{pmatrix}$$

We could solve this for a specific b vector or any one in general, lets try the general first.

```
b = {b1, b2, b3};
```

We would like the augmented matrix. This is actually not obvious how to construct by the Insert command seems to work with suitable tweaking!!

```
zz = Transpose[Insert[Transpose[a], -b, 4]]
```

$$\begin{pmatrix} 1 & 2 & 3 & -b1 \\ 7 & 5 & 6 & -b2 \\ 1 & 6 & 9 & -b3 \end{pmatrix}$$

Now, on with the show. Here is how to grab one row

```
zz[[1]]
```

{1, 2, 3, −b1}

Now use this row to eliminate the elements from the first column of the other two rows.

```
op12 = zz[[2]] - zz[[2, 1]] zz[[1]]
```

{0, −9, −15, 7 b1 − b2}

```
op13 = zz[[3]] - zz[[1]]
```

{0, 4, 6, b1 − b3}

Here is the first intermediate step matrix (see how to reconstruct a matrix).

```
mattemp1 = {zz⟦1⟧, op12, op13}
```

$$\begin{pmatrix} 1 & 2 & 3 & -b1 \\ 0 & -9 & -15 & 7\,b1 - b2 \\ 0 & 4 & 6 & b1 - b3 \end{pmatrix}$$

Now we use the second row to do the elimination. Note the necessary factor.

```
op23 = mattemp1⟦3⟧ - (mattemp1⟦3, 2⟧ mattemp1⟦2⟧) / mattemp1⟦2, 2⟧
```

$$\left\{0,\ 0,\ -\frac{2}{3},\ b1 + \frac{4}{9}(7\,b1 - b2) - b3\right\}$$

After normalizing the last two rows, for a 3 by 3 we are finished with "1"'s down the diagonal and 0's below

```
ans1 = {zz⟦1⟧, op12 / op12⟦2⟧, op23 / op23⟦3⟧}
```

$$\begin{pmatrix} 1 & 2 & 3 & -b1 \\ 0 & 1 & \frac{5}{3} & \frac{1}{9}(b2 - 7\,b1) \\ 0 & 0 & 1 & -\frac{3}{2}\left(b1 + \frac{4}{9}(7\,b1 - b2) - b3\right) \end{pmatrix}$$

The answer for {x1,x2,x3} is then obtained by back substitution.

x3=

```
x3ans = Simplify[-ans1⟦3, 4⟧]
```

$$\frac{1}{6}(37\,b1 - 4\,b2 - 9\,b3)$$

x2=

```
x2ans = Simplify[-(ans1⟦2, 3⟧ x3ans + ans1⟦2, 4⟧)]
```

$$-\frac{19\,b1}{2} + b2 + \frac{5\,b3}{2}$$

x1=

```
x1ans = Simplify[-(ans1⟦1, 2⟧ x2ans + ans1⟦1, 3⟧ x3ans + ans1⟦1, 4⟧)]
```

$$\frac{1}{2}(3\,b1 - b3)$$

We can check these with the "built in" function:

```
LinearSolve[a, b]
```

$$\left\{\frac{1}{2}(3\,b1 - b3),\ \frac{1}{2}(-19\,b1 + 2\,b2 + 5\,b3),\ \frac{1}{6}(37\,b1 - 4\,b2 - 9\,b3)\right\}$$

So it matches.

Now try a numerical example for b

Consider A x = b

```
a = {{9, 21, 3}, {4, 20, 6}, {1, 13, 9}};
```

```
b = {1, 4, 13};
```

Again we need the augmented matrix.

```
zz = Transpose[Insert[Transpose[a], -b, 4]]
```

$$\begin{pmatrix} 9 & 21 & 3 & -1 \\ 4 & 20 & 6 & -4 \\ 1 & 13 & 9 & -13 \end{pmatrix}$$

Now, on with the show. Here is how to grab one row

```
zz[[1]]
```

Again it agrees.

```
LinearSolve[a, b]
```

$$\left\{\frac{15}{16}, -\frac{11}{16}, \frac{7}{3}\right\}$$

Of course, you cannot do an elimination if the element to do the eliminating is 0. To continue you must switch rows, (partial pivoting) which does not alter anything about the solution or switch rows and columns (full pivoting) in which case you do have some book keeping to attend to. See standard numerical analysis books about this issue for more information.

■ LU decomposition

For calculational purposes, it is often convenient to express the original matrix (of course) *a*, in terms of simpler pieces that allow straightforward computation of desired properties such as inverses and also can be used for solutions where the "b" vector may have many different values.

Consider again *A* x = b
Here is the idea. We wish to create a lower triangular matrix, L and an upper triangular matrix, U such that L U = A. We will find that U is just the resulting upper triangular matrix from Gauss elimination. The Lii elements will be the "pivots" or the "on diagonal elements" before the elimination. The Lij elements are the negatives of necessary multipliers for the elimination. This makes them the matrix elements of the matrix that is being worked on at a given step. (You will probably need to read this in chapter 3 of Noble and Daniel to make sense of what is being done.)

The reason that this will be useful is that we can then solve for an arbitrary b as follows.
We create a new vector, y, by solve the forward substitution problem,

L y = b, then we solve

Ux = y

by backward substitution to get x. This idea just comes from,

L (Ux) = b with the Ux replaced by y.

If we do this, we do operations to get L and U, and then can solve for any b without doing any row operations. There is an operations count advantage if we need to solve for many different b's compared to using the augmented matrix formalizm given above.

Let's see if we can make this work. Here is what we want L to be

```
el = {{l11, 0, 0}, {l21, l22, 0}, {l31, l32, l33}}
```

$$\begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix}$$

Here is what we want U to be.

```
eu = {{1, u12, u13}, {0, 1, u23}, {0, 0, 1}}
```

$$\begin{pmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

By definition, this should (also) be A.

```
zz = el . eu
```

$$\begin{pmatrix} l_{11} & l_{11}\, u_{12} & l_{11}\, u_{13} \\ l_{21} & l_{22} + l_{21}\, u_{12} & l_{21}\, u_{13} + l_{22}\, u_{23} \\ l_{31} & l_{32} + l_{31}\, u_{12} & l_{33} + l_{31}\, u_{13} + l_{32}\, u_{23} \end{pmatrix}$$

We can see from our definition that the first column of L will be the same as the first column of A.

Now continue and check if this gives back A by undoing it with Gaussian elimination.

```
zz⟦1⟧
```

$\{l_{11}, l_{11}\, u_{12}, l_{11}\, u_{13}\}$

$$\text{op12} = \text{zz⟦2⟧} - \frac{\text{zz⟦2, 1⟧ zz⟦1⟧}}{\text{zz⟦1, 1⟧}}$$

$\{0, l_{22}, l_{22}\, u_{23}\}$

$$\text{op13} = \text{zz⟦3⟧} - \frac{\text{zz⟦3, 1⟧ zz⟦1⟧}}{\text{zz⟦1, 1⟧}}$$

$\{0, l_{32}, l_{33} + l_{32}\, u_{23}\}$

Here is the first intermediate matrix. Note the values in the second column!!

$$\text{mattemp1} = \left\{\frac{\text{zz⟦1⟧}}{\text{zz⟦1, 1⟧}}, \text{op12, op13}\right\}$$

$$\begin{pmatrix} 1 & u_{12} & u_{13} \\ 0 & l_{22} & l_{22}\, u_{23} \\ 0 & l_{32} & l_{33} + l_{32}\, u_{23} \end{pmatrix}$$

We see that we want the second column terms including and below the diagonal.

Do it again and we get the L[3,3] element.

$$\text{op23} = \text{mattemp1⟦3⟧} - \frac{\text{mattemp1⟦3, 2⟧ mattemp1⟦2⟧}}{\text{mattemp1⟦2, 2⟧}}$$

$\{0, 0, l_{33}\}$

So for U we have (which works!!) our desired upper triagonal matrix.

$$\text{ans1} = \left\{\frac{\text{zz⟦1⟧}}{\text{zz⟦1, 1⟧}}, \frac{\text{op12}}{\text{op12⟦2⟧}}, \frac{\text{op23}}{\text{op23⟦3⟧}}\right\}$$

$$\begin{pmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

Check again

```
zz - el . eu
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Now do a numerical example.

```
a = {{12, 33, 3}, {4, 20, 16}, {111, 12, 9}};
b = {1, 4, 13};
a
```

$$\begin{pmatrix} 12 & 33 & 3 \\ 4 & 20 & 16 \\ 111 & 12 & 9 \end{pmatrix}$$

```
a[[1]]
```

{12, 33, 3}

The first column of L is the same as *a*

```
l11 = a[[1, 1]]
```

12

```
l21 = a[[2, 1]]
```

4

```
l31 = a[[3, 1]]
```

111

Now perform the 2 row operations

```
op12 = a[[2]] - a[[2, 1]] a[[1]] / a[[1, 1]]
```

{0, 9, 15}

```
op13 = a[[3]] - a[[3, 1]] a[[1]] / a[[1, 1]]
```

$$\left\{0, -\frac{1173}{4}, -\frac{75}{4}\right\}$$

The intermediate matrix looks like:

```
mattemp1 = {a[[1]]/a[[1, 1]], op12, op13}
```

$$\begin{pmatrix} 1 & \frac{11}{4} & \frac{1}{4} \\ 0 & 9 & 15 \\ 0 & -\frac{1173}{4} & -\frac{75}{4} \end{pmatrix}$$

Now get the next column of L

```
l22 = mattemp1[[2, 2]]
```

9

```
l32 = mattemp1[[3, 2]]
```

$$-\frac{1173}{4}$$

We have only one row operation left

```
op23 = mattemp1[[3]] - mattemp1[[3, 2]] mattemp1[[2]] / mattemp1[[2, 2]]
```

{0, 0, 470}

Then we can get the last value for L

```
l33 = op23[[3]]
```

470

The upper matix is

```
U = {a[[1]]/a[[1, 1]], op12/op12[[2]], op23/op23[[3]]}
```

$$\begin{pmatrix} 1 & \frac{11}{4} & \frac{1}{4} \\ 0 & 1 & \frac{5}{3} \\ 0 & 0 & 1 \end{pmatrix}$$

Construct L

```
L = {{l11, 0, 0}, {l21, l22, 0}, {l31, l32, l33}}
```

$$\begin{pmatrix} 12 & 0 & 0 \\ 4 & 9 & 0 \\ 111 & -\frac{1173}{4} & 470 \end{pmatrix}$$

```
L . U
```

$$\begin{pmatrix} 12 & 33 & 3 \\ 4 & 20 & 16 \\ 111 & 12 & 9 \end{pmatrix}$$

Check against *a*

```
a
```

$$\begin{pmatrix} 12 & 33 & 3 \\ 4 & 20 & 16 \\ 111 & 12 & 9 \end{pmatrix}$$

to solve for an arbitrary b

```
b
```

{1, 4, 13}

```
y = LinearSolve[L, b]
```

$$\left\{\frac{1}{12}, \frac{11}{27}, \frac{1109}{4230}\right\}$$

```
xx = LinearSolve[U, y]
```

$$\left\{\frac{419}{4230}, -\frac{25}{846}, \frac{1109}{4230}\right\}$$

Compare this to the built-in function

```
LinearSolve[a, b]
```

$$\left\{\frac{419}{4230}, -\frac{25}{846}, \frac{1109}{4230}\right\}$$

We are glad to see that it works.

■ Inverse by Gaussian elimination

We have mentioned that we would never be calculating properties of large matrices with the direct calculation of determinants by Laplace's method. Gaussian elimination is the procedure that is most efficient at matix manipulations so we should figure out how to use it to calculate the inverse.

Since we have already seen that that for Ly = b, then, Ux=y is easily solved to get x by forward and back substitution. That is L U x = b,

$Ux = L^{-1} b = y$

$x = U^{-1} y$

which are trivial calculations as they are only forward and back substitutions. If b were composed of n vectors that represented an n identity matrix, solving for x would be the same as calculating the inverse of *a*, one column at a time. So we do this three times for the three required b's.

```
b1 = {1, 0, 0};
b2 = {0, 1, 0};
b3 = {0, 0, 1};
```

```
L
```

$$\begin{pmatrix} 12 & 0 & 0 \\ 4 & 9 & 0 \\ 111 & -\frac{1173}{4} & 470 \end{pmatrix}$$

The answer for {x1,x2,x3} is then obtained by back substitution.

x3=

$$\mathbf{y11} = \frac{\mathbf{b1[[1]]}}{\mathbf{L[[1, 1]]}}$$

$$\frac{1}{12}$$

x2=

$$\mathbf{y21} = \frac{\mathbf{b1[[2]] - y11\, L[[2, 1]]}}{\mathbf{L[[2, 2]]}}$$

$$-\frac{1}{27}$$

x1=

$$\mathbf{y31} = \frac{\mathbf{b1[[3]] - y11\, L[[3, 1]] - y21\, L[[3, 2]]}}{\mathbf{L[[3, 3]]}}$$

$$-\frac{181}{4230}$$

```
U
```

$$\begin{pmatrix} 1 & \frac{11}{4} & \frac{1}{4} \\ 0 & 1 & \frac{5}{3} \\ 0 & 0 & 1 \end{pmatrix}$$

```
x31 = y31
```

$$-\frac{181}{4230}$$

```
x21 = y21 - U[[2, 3]] x31
```

$$\frac{29}{846}$$

```
x11 = y11 - U[[1, 2]] x21 - U[[1, 3]] x31
```

$$-\frac{1}{4230}$$

```
Inverse[a]
```

$$\begin{pmatrix} -\frac{1}{4230} & -\frac{29}{5640} & \frac{13}{1410} \\ \frac{29}{846} & -\frac{5}{1128} & -\frac{1}{282} \\ -\frac{181}{4230} & \frac{391}{5640} & \frac{1}{470} \end{pmatrix}$$

So the first column checks,
Let us get the entire matrix

```
b1 = {1, 0, 0};
b2 = {0, 1, 0};
b3 = {0, 0, 1};
```

define the identity matrix

```
bi = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}};
```

initialize the y matrix

```
y = {{1, 1, 1}, {1, 1, 1}, {1, 1, 1}}
```

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

```
L
```

$$\begin{pmatrix} 12 & 0 & 0 \\ 4 & 9 & 0 \\ 111 & -\frac{1173}{4} & 470 \end{pmatrix}$$

We can easily solve the forward substitution (the value of this is to write it in a *Mathematica* loop)

```
Do[y[[1, i]] = bi[[i, 1]]/L[[1, 1]]; y[[2, i]] = (bi[[i, 2]] - y[[1, i]] L[[2, 1]])/L[[2, 2]];
 y[[3, i]] = (bi[[i, 3]] - y[[1, i]] L[[3, 1]] - y[[2, i]] L[[3, 2]])/L[[3, 3]];, {i, 1, 3}];
```

Which gives

```
y
```

$$\begin{pmatrix} \frac{1}{12} & 0 & 0 \\ -\frac{1}{27} & \frac{1}{9} & 0 \\ -\frac{181}{4230} & \frac{391}{5640} & \frac{1}{470} \end{pmatrix}$$

```
U
```

$$\begin{pmatrix} 1 & \frac{11}{4} & \frac{1}{4} \\ 0 & 1 & \frac{5}{3} \\ 0 & 0 & 1 \end{pmatrix}$$

We can check this one

```
LinearSolve[U, y]
```

$$\begin{pmatrix} -\frac{1}{4230} & -\frac{29}{5640} & \frac{13}{1410} \\ \frac{29}{846} & -\frac{5}{1128} & -\frac{1}{282} \\ -\frac{181}{4230} & \frac{391}{5640} & \frac{1}{470} \end{pmatrix}$$

```
y
```

$$\begin{pmatrix} \frac{1}{12} & 0 & 0 \\ -\frac{1}{27} & \frac{1}{9} & 0 \\ -\frac{181}{4230} & \frac{391}{5640} & \frac{1}{470} \end{pmatrix}$$

Now the back substitution

```
ainv = {{1, 1, 1}, {1, 1, 1}, {1, 1, 1}};
```

```
Do[ainv[[3, i]] = y[[3, i]]; ainv[[2, i]] = y[[2, i]] - ainv[[3, i]] U[[2, 3]];
 ainv[[1, i]] = y[[1, i]] - ainv[[2, i]] U[[1, 2]] - ainv[[3, i]] U[[1, 3]];,
 {i, 1, 3}];
```

```
ainv
```

$$\begin{pmatrix} -\frac{1}{4230} & -\frac{29}{5640} & \frac{13}{1410} \\ \frac{29}{846} & -\frac{5}{1128} & -\frac{1}{282} \\ -\frac{181}{4230} & \frac{391}{5640} & \frac{1}{470} \end{pmatrix}$$

Which checks with the built in function.

```
Inverse[a]
```

$$\begin{pmatrix} -\frac{1}{4230} & -\frac{29}{5640} & \frac{13}{1410} \\ \frac{29}{846} & -\frac{5}{1128} & -\frac{1}{282} \\ -\frac{181}{4230} & \frac{391}{5640} & \frac{1}{470} \end{pmatrix}$$

The LU factorization can be done with a canned package also, but it employs partial pivoting so there can be differences in the LU

Here we load the package:

```
Needs["LinearAlgebra`GaussianElimination`"]
```

```
LUFactor[a]
```

$$\mathrm{LU}\left(\begin{pmatrix} \frac{4}{37} & \frac{1173}{37} & \frac{75}{37} \\ \frac{4}{111} & \frac{724}{1173} & \frac{5640}{391} \\ 111 & 12 & 9 \end{pmatrix}, \{3, 1, 2\}\right)$$

Solve for the inverse one column at a time:

```
b1 = {1, 0, 0};
b2 = {0, 1, 0};
b3 = {0, 0, 1};
```

```
row1 = LUSolve[LUFactor[a], b1]
```

$$\left\{-\frac{1}{4230}, \frac{29}{846}, -\frac{181}{4230}\right\}$$

```
row2 = LUSolve[LUFactor[a], b2]
```

$$\left\{-\frac{29}{5640}, -\frac{5}{1128}, \frac{391}{5640}\right\}$$

```
row3 = LUSolve[LUFactor[a], b3]
```

$$\left\{\frac{13}{1410}, -\frac{1}{282}, \frac{1}{470}\right\}$$

```
temp = {row1, row2, row3}
```

$$\begin{pmatrix} -\frac{1}{4230} & \frac{29}{846} & -\frac{181}{4230} \\ -\frac{29}{5640} & -\frac{5}{1128} & \frac{391}{5640} \\ \frac{13}{1410} & -\frac{1}{282} & \frac{1}{470} \end{pmatrix}$$

```
inversea = Transpose[temp]
```

$$\begin{pmatrix} -\frac{1}{4230} & -\frac{29}{5640} & \frac{13}{1410} \\ \frac{29}{846} & -\frac{5}{1128} & -\frac{1}{282} \\ -\frac{181}{4230} & \frac{391}{5640} & \frac{1}{470} \end{pmatrix}$$

```
Inverse[a]
```

$$\begin{pmatrix} -\frac{1}{4230} & -\frac{29}{5640} & \frac{13}{1410} \\ \frac{29}{846} & -\frac{5}{1128} & -\frac{1}{282} \\ -\frac{181}{4230} & \frac{391}{5640} & \frac{1}{470} \end{pmatrix}$$

So we get the same answer!!