THE UNIVERSITY OF SYDNEY

MATH2008 Introduction to Modern Algebra

(http://www.maths.usyd.edu.au/u/UG/IM/MATH2008/)

Semester2, 2003 — Lecturer: R. Howlett

Computer Tutorial 3

The purpose of this tutorial to learn how to use MAGMA in computations with subspaces of vector spaces, how to use MAGMA to solve matrix equations, and how to use matrices to calculate projections.

The main MAGMA commands in this tutorial are:

`sub`   `KMatrixSpace`   `Transpose`   `Solution`   `NullSpace`

It is recommended that you set a log file at the start of each MAGMA session to keep a record of your work. e.g. `SetLogFile("ctut3.mlg");`. (Don't forget the semicolon!)

To save you some typing, a file called `t3defs.m` has been created containing the following MAGMA code:

```
Length := func< u | Sqrt(InnerProduct(u,u)) >;
Angle  := func< u,v | Arccos(InnerProduct(u,v)
                      /(Length(u)*Length(v))) >;
asDegree   := func< x | x*180/Pi(RealField()) >;
Projection := func< Q, v | Solution(A*B,v*B)*A
            where B is Transpose(A)
            where A is BasisMatrix(Q) >;
```

To get MAGMA to read this file, use the command `load "t3defs.m";`.(Incidentally, you can do this kind of thing for yourself. If you create a file containing lines of MAGMA commands, called `MyDefinitions.m` (say), you can get MAGMA to execute these commands by typing `load "MyDefinitions.m";` at the MAGMA prompt. You can use NEDIT to create such a file.)

If `V` is a vector space, and `a`, `b`, `c` (etc.) are vectors in `V`, then (for example) the command `W:=sub< V | a,b,c >;` creates the subspace of `V` spanned by `a`, `b`, and `c`.

1. (*i*) Apply the Gram-Schmidt process to the vectors

$$(1,1,1,0,0),\ (1,0,-1,0,-1),\ (2,-1,1,1,0),\ (4,0,0,1,-1).$$

(First define `V` as a vector space of dimension 5 over the real field and enter the above vectors as `a1` to `a4`. Then get MAGMA to compute `v1` to `v4`, where `v1 = a1`, and for `i` $>$ 1 the vector `vi` is `ai` minus the projection of `ai` onto the subspace spanned by the earlier `v`'s. You will need to use MAGMA commands like `v3:=a3 - Projection(sub< V | v1,v2 >, a3);`.)

(*ii*) Convert the orthogonal set you found in Part (*i*) into an orthonormal one.

Solution.

```
> R := RealField();
> V := VectorSpace(R,5);
> a1 := V![1,1,1,0,0];
> a2 := V![1,0,-1,0,-1];
> a3 := V![2,-1,1,1,0];
> a4 := V![4,0,0,1,-1];
> u1 := a1;
> u2 := a2 - Projection( sub< V | u1 >, a2 );
> u3 := a3 - Projection( sub< V | u1,u2 >, a3 );
> u4 := a4 - Projection( sub< V | u1,u2,u3 >, a4 );
> print u1, u2, u3, u4;
(1 1 1 0 0)
( 1 0 -1 0 -1)
( 1 -5/3 2/3 1 1/3)
( 1/8 1/8 -1/4 1/8 3/8)
> u1 := u1/Length(u1);
> u2 := u1/Length(u2);
> u3 := u1/Length(u3);
> u4 := u1/Length(u4);
> print u1;
> (0.57735.. ,0.57735.. (etc.)
```

2. Find an orthonormal set of vectors by applying the Gram-Schmidt process to

$$(1,0,0,0,0,0),\ (1,1,0,0,0,0),\ (1,1,1,0,0,0),\ (1,1,1,1,0,0).$$

Solution.

The answer is $(1,0,0,0,0,0)$, $(0,1,0,0,0,0)$, $(0,0,1,0,0,0)$, $(0,0,0,1,0,0)$; these vectors already have length 1, as required for an orthonormal basis.

```
> R := RealField();
> V := VectorSpace(R,6);
> a1 := V![1,0,0,0,0,0];
> a2 := V![1,1,0,0,0,0];
> a3 := V![1,1,1,0,0,0];
> a4 := V![1,1,1,1,0,0];
> u1 := a1;
> u2 := a2 - Projection( sub< V | u1 >, a2 );
> u3 := a3 - Projection( sub< V | u1,u2 >, a3 );
```

```
> u4 := a4 - Projection( sub< V | u1,u2,u3 >, a4 );
> u1, u2, u3, u4;
(1 0 0 0 0 0)
(0 1 0 0 0 0)
(0 0 1 0 0 0)
(0 0 0 1 0 0)
> Length(u1);
1.000000000000000000000000000000000000000
> Length(u2);
1.000000000000000000000000000000000000000
> Length(u3);
1.000000000000000000000000000000000000000
> Length(u4);
1.000000000000000000000000000000000000000
```

3. The nullspace of an $m \times n$ matrix A is usually defined to be the space of all n-component column vectors $\underset{\sim}{v}$ such that $A\underset{\sim}{v} = \underset{\sim}{0}$. Since MAGMA uses row vectors, the MAGMA function `NullSpace` returns the space of all m-component row vectors $\underset{\sim}{v}$ such that $\underset{\sim}{v}A = 0$. We shall call this the *left nullspace* of A, and the usual nullspace of A the *right nullspace* of A. Use MAGMA to find the left nullspace of

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

How can you use MAGMA to find the right nullspace of the same matrix?

Solution.

```
> M32 := KMatrixSpace(R,3,2);
> S := M32![1,-1,2,-2,5,-5];
> NullSpace(S);
Vector space of degree 3, dimension 2 over Real Field
Echelonized basis:
( 1 0 -1/5)
( 0 1 -2/5)
> // This gives the left nullspace.  The right nullspace is the
> // transpose of the left nullspace of the transpose of S.
> NullSpace(Transpose(S));
Vector space of degree 2, dimension 1 over Real Field
Echelonized basis:
(1 1)
```

So the right nullspace of `S` has dimension 1 and is spanned by $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

I hope that you can see immediately that this is true without using MAGMA (or even a pen and paper).

4. (i) Find the projection of $\underset{\sim}{v} = (3, -2, 4)$ onto $W = \{\, (x, y, z) \mid 2x - y + 3z = 0 \,\}$, a subspace of $(\mathbb{R}^3)'$.

(Observe that W consists of the solutions of the equation

$$(x, y, z)\begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix} = 0.$$

The sequence of MAGMA commands

```
M := KMatrixSpace(RealField(),3,1);
A := M![2,-1,3];
W := NullSpace(A);
```

will create W for you.)

(ii) Geometrically, what is the space W from Part (i)? (This is not a computer question!)

(iii) Find an orthogonal matrix whose first row is $\frac{1}{\sqrt{14}}(2, -1, 3)$. (Hint. The 2nd and 3rd rows must form an orthonormal basis for the space W in Part (i).)

Solution.

We have already declared `R` to be the real field in our answer to Exercise 1. So we do not have to do that now.

For Part (i):

```
> V := VectorSpace(R,3);
> v := V![3,-2,4];
> M := KMatrixSpace(R,3,1);
> A := M![2,-1,3];
> W := NullSpace(A);
> W;
Vector space of degree 3, dimension 2 over Real Field
Echelonized basis:
( 1 0 -2/3)
( 0 1 1/3)
> Projection(W,v);
( 1/7 -4/7 -2/7)
```

(ii) The equation $2x - y + 3z = 0$ defines a plane in $\mathbb{R}^3$. Observe that $2x-y+3z$ is the dot product of the vectors $\underset{\sim}{x} = (x, y, z)^T$ and $\underset{\sim}{u} = (2, -1, 3)^T$. Thus point whose position vector (relative to the origin) is $\underset{\sim}{x}$ lies in the plane $2x - y + 3z = 0$ if and only if $\underset{\sim}{x} \cdot \underset{\sim}{u} = 0$; that is, if and only if $\underset{\sim}{x}$ is perpendicular to $\underset{\sim}{u}$. So the space W is the plane through the origin orthogonal to $(2, -1, 3)^T$.

(iii) Observe that if $\underset{\sim}{u} = (2, -1, 3)$, then $\|\underset{\sim}{u}\| = \sqrt{2^2 + 1^2 + 3^2} = \sqrt{14}$. So the given vector is $(1/\|\underset{\sim}{u}\|)\underset{\sim}{u}$, the unit vector in the direction of $\underset{\sim}{u}$. The other rows of the desired matrix must therefore be orthogonal to $\underset{\sim}{u}$, and hence in the subspace W. Indeed, they must form an orthonormal basis of W. Moreover, any orthonormal basis of W will do. So we can get a solution by applying the Gram-Schmidt process to the echelonized basis of W that MAGMA has already found for us, and then normalizing the resulting basis vectors.

```
> V := VectorSpace(R,3);
> u := V![2,-1,3];
> w1 := V![1,0,-2/3];
> w2 := V![0,1,1/3];
> w2 := w2-(InnerProduct(w2,w1)/InnerProduct(w1,w1))*w1;
> u := u/Length(u);
> w1 := w1/Length(w1);
> w2 := w2/Length(w2);
> MM := KMatrixSpace(R,3,3);
> A := MM![u,w1,w2];
> A;
> [0.534522483824848769369106961759507043105
> -0.267261241912424384684553480879753521552
> 0.801783725737273154053660442639260564658]
> [0.832050294337843683027512600185499064521 0.E-38
> -0.554700196225229122018341733456999376343]
> [0.148249863332220235885168193532918076618 8
> 0.963624111659431533253593257963967498032347358133049220644 6
> 0.222374794998330353827752290299377114932211755600585176565 4]
```

This is a good approximation to the correct solution

$$A = \begin{pmatrix} \frac{2}{\sqrt{14}} & \frac{-1}{\sqrt{14}} & \frac{3}{\sqrt{14}} \\ \frac{3}{\sqrt{13}} & 0 & \frac{-2}{\sqrt{13}} \\ \frac{2}{\sqrt{182}} & \frac{\sqrt{13}}{\sqrt{14}} & \frac{3}{\sqrt{182}} \end{pmatrix}$$

There are many other correct answers, since the space W has many orthonormal bases.

5. (Harder) In this exercise we shall use MAGMA to construct a function called `orthog`. When given a row vector `v`, the `orthog` function will return the space of row vectors orthogonal to `v`.

First of all, the function is going to look something like

```
orthog := func< v | W >;
```

where W is the subspace we want.

We know that a row vector $\underset{\sim}{x}$ is orthogonal to the row vector $\underset{\sim}{v}$ if and only if $\underset{\sim}{x} \cdot \underset{\sim}{v} = 0$. In matrix terms this can be written as $\underset{\sim}{x}\underset{\sim}{v}^T = 0$. In other words the vectors $\underset{\sim}{x}$ form the left nullspace of $\underset{\sim}{v}^T$ (which is a column vector regarded as a matrix). Thus in MAGMA we could try to define W as `NullSpace(Transpose(v))`. This doesn't work because MAGMA will know

that `v` is a vector, but it can only transpose matrices, and does not regard vectors as matrices. Therefore we have to coerce MAGMA into thinking of `v` as a $1 \times n$ matrix, where n is the dimension of the vector space in which `v` lives. If `M` is the space of $1 \times n$ matrices we can get what we want by writing `NullSpace(Transpose(M!v))`. Thus the `orthog` function will actually look like

```
orthog := func< v | NullSpace(Transpose(M!v)) >;
```

If we type this in and try it out, MAGMA will complain and tell us that the identifier `M` has not been declared. That is because we haven't yet told it about `M`. We can fix this by adding a `where` clause of the form

```
where M is KMatrixSpace(R,1,n)
```

This is all very well, but now what is `R`, and how do we tell MAGMA what `n` is? Well, if `V` is the vector space in which `v` lives, `R` is its field of scalars and `n` is its dimension. So the `where` clause should really be

```
where M is KMatrixSpace(Field(V),1,Dimension(V))
```

But we need yet another `where` clause to tell MAGMA about `V`. This is easy because `V` is the vector space that contains `v`, and MAGMA calls this the *parent* vector space of `v`. So what we have to say is

```
where V is Parent(v)
```

Putting all this together our function becomes

```
orthog := func< v | NullSpace(Transpose(M!v))
    where M is KMatrixSpace(Field(V),1,Dimension(V))
    where V is Parent(v) >;
```

To test this out, try it on the vector $(2, -1, 3)$ used in Exercise 4.

Solution.

```
> orthog := func< v | NullSpace(Transpose(M!v))
>     where M is KMatrixSpace(Field(V),1,Dimension(V))
>     where V is Parent(v) >;
> V := VectorSpace(R,3);
> orthog(V![2,-1,3]);
Vector space of degree 3, dimension 2 over Real Field
Echelonized basis:
( 1 0 -2/3)
( 0 1 1/3)
```