Computer Tutorial 3

The purpose of this tutorial is 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` for creating subspaces of vector spaces
- `Transpose` for transposing matrices
- `Solution` for solving matrix equations
- `NullSpace` for finding nullspaces

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:

```magma
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 `v` 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.

   ```magma
   > 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), (1,1,0,0,0), (1,1,1,0,0), (1,1,1,1,0).

   Solution.

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

```magma
> 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 );
```
3. The nullspace of an \( m \times n \) matrix \( A \) is usually defined to be the space of all \( n \)-component column vectors \( v \) such that \( Av = 0 \). Since MAGMA uses row vectors, the MAGMA function \texttt{NullSpace} returns the space of all \( m \)-component row vectors \( v \) such that \( vA = 0 \). We shall call this the \textit{left nullspace} of \( A \), and the usual nullspace of \( A \) the \textit{right nullspace} of \( A \). Use MAGMA to find the left nullspace of
\[
\begin{pmatrix}
1 & -1 \\
2 & -2 \\
5 & -5
\end{pmatrix}
\]

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):
\[
> \texttt{V := VectorSpace(R,3);} \\
> \texttt{v := V! [3,-2,4];} \\
> \texttt{M := KMatrixSpace(R,3,1);} \\
> \texttt{A := M! [2,-1,3];} \\
> \texttt{W := NullSpace(A);} \\
> \texttt{W;} \\
\]
Vector space of degree 3, dimension 2 over Real Field
Echelonized basis:
\[
\begin{pmatrix}
1 & -1 \\
2 & -2 \\
5 & -5
\end{pmatrix}
\]

(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 do not have to do that now.

For Part (i):
\[
> \texttt{M32 := KMatrixSpace(R,3,2);} \\
> \texttt{S := M32! [1,-1,2,-2,5,-5];} \\
> \texttt{NullSpace(S);} \\
\]
Vector space of degree 3, dimension 2 over Real Field
Echelonized basis:
\[
\begin{pmatrix}
1 & -1 \\
2 & -2 \\
5 & -5
\end{pmatrix}
\]

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).
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.

```maple
V := VectorSpace(R,3);
u := V![2,-1,3];
w1 := V![1,0,-2/3];
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;
```

This is a good approximation to the correct solution

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

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

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

where $W$ is the subspace we want.

We know that a row vector $x$ is orthogonal to the row vector $v$ if and only if $x \cdot v = 0$. In matrix terms this can be written as $x v^T = 0$. In other words the vectors $x$ form the left nullspace of $v^T$ (which is a column vector regarded as a matrix). Thus in MAGMA we could try to define $W$ as NullSpace(Transpose($v$)). Thus the orthog function will actually look like

```maple
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

```maple
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

```maple
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

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

Putting all this together our function becomes

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

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

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

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

Solution.

```maple
V := VectorSpace(R,3);
A := Orthog(V[2,-1,3]);
A;
```

Vector space of degree 3, dimension 2 over Real Field

Echelonized basis:

```
( 1 0 -2/3)
( 0 1 1/3)
```