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 1

In this tutorial you will use MAGMA to calculate the length of vectors and the inner products and distances between vectors. In the course of this you will be introduced to the following MAGMA commands:

`RealField` `VectorSpace` `print` `func` `InnerProduct` `Sqrt` `Arccos`

First, start MAGMA and type

`SetLogFile("ctut1.mlg");`

Near the end of the tutorial, select "NEDIT" from the applications menu of Session Manager and open the file ctut1.mlg. You will see that it contains a record of your magma session.

When you have finished the tutorial, exit from magma by typing `quit;`, and then logout by selecting "logout" in Session Manager.

1. *Setting up the vector space.* Use MAGMA to create a vector space `V` of dimension 4 over the real numbers `R`. Then define two vectors $(4, 3, 2, 1)$ and $(1, -2, 0, -3)$ named `a` and `b`. The commands are

```
R := RealField();
V := VectorSpace(R,4);
a := V![4,3,2,1];
b := V![1,-2,0,-3];
```

Points to note:

(a) Each statement ends with a semicolon (`;`). If you press return before you type `;` , simply type it on the next line.

(b) You use `:=` to assign values.

(c) MAGMA treats vectors as *row* vectors. Also, you need to tell MAGMA which vector space will contain the vector. This is why you use `V!` before each vector. If you had another vector space, called `mySpace` and of dimension 3, then you would use the command `z := mySpace![1,2,3]` to create the vector $(1, 2, 3)$ named `z` in the vector space `mySpace`.

Solution.

```
> R := RealField();
> V := VectorSpace(R,4);
> a := V![4,3,2,1];
> b := V![1,-2,0,-3];
```


2. *Displaying values.* To see the value of the quantities you have defined, you can use the `print` command. For example

```
print a, b;
print V;
```

In fact the word `print` can be omitted if you wish.

Solution.

```
> a, b;
(4 3 2 1)
( 1 -2 0 -3)
> V;
Full Vector space of degree 4 over Real Field
```

3. *Inner products and length.* You can get these with the following commands:

```
InnerProduct(a,b);
Sqrt(InnerProduct(a,a));
```

(or `print InnerProduct(a,b)` and `print Sqrt(InnerProduct(a,a))`)

Solution.

```
> print InnerProduct(a,b);
-5
> print Sqrt(InnerProduct(a,a));
5.477225575051661134569697828006
```

4. *Angles.* You should be able to find the angle between `a` and `b` using the commands from the previous exercise together with the `Arccos` function. Try this now. You will see that this involves quite a lot of typing and very often you will make typing mistakes. To make things easier, first define an abbreviation for the `Length` function

```
Length := func< v | Sqrt(InnerProduct(v,v)) >;
```

and then define the `Angle` function as

```
Angle := func< u,v | Arccos(InnerProduct(u,v)
/ (Length(u)* Length(v))) >;
```

Now you can print the angle between `a` and `b` using

```
print Angle(a,b);
```

(Notice that `*` is used for multiplication.) The answer will be in radians. But what if you want the answer in degrees? Here is one way to get it:

```
asDegree := func< x | x * 180 /Pi(R) >;
print asDegree(Angle(a,b));
```

Notice that in MAGMA you use `Pi(R)` to get the number π. (The `R` in this refers to the real field `R` (to which π belongs). MAGMA always needs to be told which set contains each object.)

Solution.

```
> Length := func< v | Sqrt(InnerProduct(v,v)) >;
> Angle := func< u,v | Arccos(InnerProduct(u,v) /
(Length(u) * Length(v))) >;
> print Angle(a,b);
1.81725895828755442085570281935
> asDegree := func< x | x * 180/Pi(R) >;
> print asDegree(Angle(a,b));
104.121268592217380500152290452277
```

5. Exercises 6 and 7 below are the same as Exercises 1 and 2 on Tutorial Sheet 1. Hopefully MAGMA will give the same answers as you will get with with pen and paper. To prepare for these exercises, define a MAGMA function `Distance` such that `Distance(a,b)` gives the distance from `a` to `b`.

Solution.

```
> Distance := func< u,v | Length(u-v) >;
> Distance(a,b);
7.348469228349534294591852224059
```

6. Let $\underset{\sim}{u} = (1, 0, -5, 7)$ and $\underset{\sim}{v} = (21, 2, 2, -2)$. Find the lengths of $\underset{\sim}{u}$ and $\underset{\sim}{v}$ and the angle between them.

Solution.

```
> u := V![1,0,-5,7];
> v := V![21,2,2,-2];
> print Length(u);
8.66025403784438646763723170749
> print Length(v);
21.283796653792762638652980918205
> print Angle(u,v);
1.58707281463299704056243849274
> print asDegree(Angle(u,v));
90.93257405841916767609009713042
```

7. The four points $(0, 0, 0)$, $(1, 1, 0)$, $(1, 0, 1)$ and $(0, 1, 1)$ are the vertices of a tetrahedron in $\mathbb{R}^3$.

(i) Show that all six edges of this tetrahedron have the same length.

(ii) Given that the centre of this tetrahedron is $(1/2, 1/2, 1/2)$, calculate the angle between two rays joining the centre to two of the vertices. Check that you get the same answer whichever two vertices you choose.

(This tetrahedron can be seen as a model of a methane molecule, with a carbon atom at the centre and hydrogen atoms at the vertices. The angle in Part (ii) is the "bond angle".)

Solution.

```
> W := VectorSpace(R,3);
> u1 := W![0,0,0]; > u2 := W![1,1,0];
> u3 := W![1,0,1];
> u4 := W![0,1,1];
> print Distance(u1,u2);
1.414213562373095048801688724198
> print Distance(u1,u3);
1.414213562373095048801688724198
> print Distance(u1,u4);
1.414213562373095048801688724198
> print Distance(u2,u3);
1.414213562373095048801688724198
> print Distance(u2,u4);
1.414213562373095048801688724198
> print Distance(u3,u4);
1.414213562373095048801688724198
> c := W![1/2,1/2,1/2];
> r1 := u1 - c;
> r2 := u2 - c;
> r3 := u3 - c;
> r4 := u4 - c;
> print Angle(r1,r2);
1.91063323624901855632771420499
> print Angle(r1,r3);
1.91063323624901855632771420499
> print Angle(r1,r4);
1.91063323624901855632771420499
> print Angle(r2,r3);
1.91063323624901855632771420499
> print Angle(r2,r4);
1.91063323624901855632771420499
> print Angle(r3,r4);
1.91063323624901855632771420499
> print asDegree(Angle(r1,r2));
109.471220634490691369245999336544
> print asDegree(Angle(r1,r3));
109.471220634490691369245999336544
> print asDegree(Angle(r1,r4));
109.471220634490691369245999336544
> print asDegree(Angle(r2,r3));
109.471220634490691369245999336544
> print asDegree(Angle(r2,r4));
109.471220634490691369245999336544
> print asDegree(Angle(r3,r4));
109.471220634490691369245999336544
> quit;
```