Worksheet 2

Version 2014-12-03

2 Basic linear algebra with GAP

2.1 Vectors

In GAP, a typical way to represent vectors is as lists of coefficients, i.e. as coordinate vectors.

```
gap> v := [1,2,3];
[ 1, 2, 3 ]
```

This is quite efficient and already allows for typical operations, like scalar multiplication and vector addition. GAP even allows you to "multiply" vectors, which computes the standard scalar product:

```
gap> w := [-3,0,1];
[ -3, 0, 1 ]
gap> 3 * v + w;
[ 0, 6, 10 ]
gap> v * [1,1,1];
6
```

2.2 Matrices

In GAP, a matrix typically is represented as a list of row vectors. That is, a list of lists.

```
gap> id := [[1,0,0], [0,1,0], [0,0,1]];
[ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ]
gap> Display(id); # Print a nice view
[ [  1,  0,  0 ],
  [  0,  1,  0 ],
  [  0,  0,  1 ] ]
```

You can add and multiply matrices as you would expect

```
gap> m := [ [ -2, 3, 2 ], [ 0, 0, 1 ], [ 5, -7, -5 ] ];;
gap> Display(m * m);
[ [   14,  -20,  -11 ],
  [    5,   -7,   -5 ],
  [  -35,   50,   28 ] ]
gap> 3 * m + m;
[ [ -8, 12, 8 ], [ 0, 0, 4 ], [ 20, -28, -20 ] ]
```

Here is an example where we multiply a non-square matrix by its transpose in two different ways:

```
gap> a := [[2,1], [0,1], [-1,0]];; Display(a);
[ [   2,   1 ],
  [   0,   1 ],
  [  -1,   0 ] ]
gap> b := TransposedMat(a);; Display(b);
[ [   2,   0,  -1 ],
  [   1,   1,   0 ] ]
gap> Display(a * b);
[ [   5,   1,  -2 ],
  [   1,   1,   0 ],
  [  -2,   0,   1 ] ]
gap> Display(b * a);
[ [  5,  2 ],
  [  2,  2 ] ]
```

Since matrices are lists of lists, you can access their entries and determine their dimensions as for lists. This matches our usual notation, where e.g. m_{31} indicates the first entry of the third row.

```
gap> m := [ [ -2, 2 ], [ 0, 1 ], [ 5, -7 ] ];;
gap> m[3]; # 3rd row
[ 5, -7 ]
gap> m[3][1]; # 3rd row, 1st entry
5
gap> Length(m); # number of rows
3
gap> Length(m[1]); # number of columns
2
gap> DimensionsMat(m); # a more expressive way of computing the matrix size
[ 3, 2 ]
```

Exercise 2.2.1

Write function `trace(m)` which takes a matrix m and computes its trace. Try to write it so that it works for non-square matrices, too.

Exercise 2.2.2

Write a function `AddMat(a,b)` which takes two matrices a, b, and returns either their sum, or `fail` (if they cannot be added). Do not use $a + b$, instead use loops and operations on matrix entries only.

For square matrices, you can compute their determinant, invert them, and more.

```
gap> m := [ [ -2, 3, 2 ], [ 0, 0, 1 ], [ 5, -7, -5 ] ];;
gap> Determinant(m);
1
gap> Display(m^-1);
[ [  7,  1,  3 ],
  [  5,  0,  2 ],
  [  0,  1,  0 ] ]
```

Many other handy functions can be found in Chapter 24 of the GAP reference manual. Some examples:

```
gap> IdentityMat(3); # 3x3 identity matrix
[ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ]
gap> NullMat(2,3); # 2x3 zero matrix
[ [ 0, 0, 0 ], [ 0, 0, 0 ] ]
gap> TraceMat([[1,2],[3,4]]);
5
```

2.3 Vector spaces and bases

Using lists of coordinates, while quite efficient from a technical point of view, lacks from a mathematical point of view. For example, it is not clear over which field we are operating. In the example above, it could be $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ or something else. Also, in applications often one needs to deal with arbitrary bases. For this, GAP allows creating objects describing vector spaces, bases and so on. Here are just some examples, for details please have a look at Chapter 61 of the GAP reference manual.

```
gap> V := Rationals^3;
( Rationals^3 )
gap> Basis(V);
CanonicalBasis( ( Rationals^3 ) )
gap> BasisVectors(Basis(V));
[ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ]
gap> B := Basis(V, [ [1,2,3], [-3,0,1], [1,1,1] ]);
Basis( ( Rationals^3 ), [ [ 1, 2, 3 ], [ -3, 0, 1 ], [ 1, 1, 1 ] ] )
```

Exercise 2.3.1

What happens if you give vectors to `Basis` which do not form a basis, or even are of the wrong dimension? Experiment!

We can ask GAP to compute coordinate vectors with respect to a certain basis via the `Coefficients` command, and do the converse via `LinearCombination`.

```
gap> v1 := [1,2,3];; v2 := [-3,0,1];; v3 := [1,1,1];;
gap> B := Basis(V, [v1, v2, v3]);;
gap> Coefficients(B, [2,1,0]);
[ -1, 0, 3 ]
gap> -1 * v1 + 0 * v2 + 3 * v3; # Let's double-check
[ 2, 1, 0 ]
gap> LinearCombination(B, [-1,0,3]);
[ 2, 1, 0 ]
```

2.4 Left and right multiplication

Let $\mathbb{K}$ be a field, and $A \in \mathbb{K}^{n\times m} = M_{n,m}(\mathbb{K})$ be an $n \times m$ matrix over $\mathbb{K}$. Then

$$\lambda_A : \mathbb{K}^m \to \mathbb{K}^n : v \mapsto Av \qquad \text{and} \qquad \rho_A : \mathbb{K}^{1\times n} \to \mathbb{K}^{1\times m} : v \mapsto vA$$

both are linear maps. The former is called left multiplication, the latter right multiplication.

Exercise 2.4.1

Implement two functions `LeftMul(A,v)` and `RightMul(A,v)` which compute left and right multiplication of a matrix A and a vector v. Return fail if Av resp. vA cannot be computed. Do not use *, rather use loops to compute the result.

As we see, given the matrix A, it is not quite clear which of the two linear maps it represents. In my Linear Algebra class, we used left multiplication. But in GAP, the standard is to consider right multiplication.

This has many consequences. For example: If V is a vector space and $\varphi : V \to V$ is a linear map, then $v \in V \setminus \{0\}$ is an eigenvector with eigenvalue $\lambda \in \mathbb{K}$ if $\varphi(v) = \lambda v$. But what then is the eigenvector of a matrix? It depends on whether you consider left or right multiplication:

Let $A \in M_n(\mathbb{K})$. Then v is a left eigenvector of A with eigenvalue λ if $vA = \lambda v$, and w is a right eigenvector of A with eigenvalue λ if $Aw = \lambda w$. Somewhat confusingly, when considering left multiplication, one is interested in right eigenvectors, and vice versa.[1]

```
gap> A:=[[1,1],[0,1]];; Display(A);
[ [  1,  1 ],
  [  0,  1 ] ]
gap> A * [1,0] = [1,0]; # [1,0] is a right eigenvector
true
gap> [1,0] * A = [1,0]; # [1,0] is not a left eigenvector
false
gap> [0,1] * A = [0,1]; # [0,1] is a left eigenvector
true
gap> Eigenvectors(Rationals, A); # GAP computes left eigenvectors
[ [ 0, 1 ] ]
```

Exercise 2.4.2

Let $A \in M_n(\mathbb{K})$.

1. Let $\tau : \mathbb{K}^n \to \mathbb{K}^{1\times n}$ be the canonical isomorphism mapping n-dimensional column vectors to row vectors. Show that $\tau \circ \lambda_A = \rho_{A^T} \circ \tau$.
2. Prove that the sets of left and right eigenvalues of A are equal.
3. Write a *short* function `RightEigenvectors(F,A)` which uses the GAP functions `Eigenvectors` and `TransposedMat` to compute right eigenvectors of a matrix A. Test it.

2.5 Homework

(s2-h1) Write a function `TransMat(a)` which takes a matrix a and returns its transpose a^T, without modifying a. As always, do not use higher GAP functions like `TransposedMat`.

(s2-h2) Implement a function `MulMat(a,b)` which performs matrix multiplication; that is, it takes two matrices a and b, and returns a new matrix c equal to ab, but without using GAP's built-in functionality for matrix multiplication. If a and b cannot be multiplied, return `fail`.

(s2-h3) Write a function `BaseChangeMat(B,C)` which takes two vector space bases $B = (b_1, \ldots, b_n)$ and $C = (c_1, \ldots, c_n)$ and computes the base change matrix $M_C^B(\mathrm{id}) = (\gamma_C(b_1) \cdots \gamma_C(b_n))$.

[1] To me, this is another example where working without bases, i.e. with linear maps instead of matrices, is actually easier.