Sage and Linear Algebra Worksheet

FCLA Section VR

Robert Beezer
Department of Mathematics and Computer Science
University of Puget Sound

Spring 2017

1 Vector Representations

It is easy to form vector representations of vectors in $\mathbb{C}^n$.

We get a nonstandard basis quickly from the columns of a nonsingular matrix. The keyword `algorithm='unimodular'` requests a matrix with determinant 1.

```
n = 6
A = random_matrix(QQ, n, algorithm='unimodular',
    upper_bound=9)
A
```

The columns of `A` become the "user basis" of a vector space.

```
B = A.columns()
V = (QQ^n).subspace_with_basis(B)
V
```

```
u = random_vector(QQ, n)
u
```

Now, we get values of the invertible linear transformation ρ_B with the Sage method `.coordinate_vector()` method of the vector space.

```
c = V.coordinate_vector(u)
c
```

The inverse linear transformation is also available as the `.linear_combination_of_basis()` method of the vector space.

```
round_trip = V.linear_combination_of_basis(c)
round_trip
```

And the automated check:

```
u == round_trip
```

Notice that this is something we could do "by hand" with just reduced row-echelon form. The coordinitization of `u` relative to the basis `B` is just a (unique) solution to a linear system.

```
aug = column_matrix(B + [u])
aug.rref()
```

The following stanza will always return `True` as we "coordinatize" and then use the coordinates to form a linear combination of the basis.

```
w = random_vector(QQ, n)
x = V.coordinate_vector(w)
y = V.linear_combination_of_basis(x)
y == w
```

2 Abstract Vector Spaces

Sage does not implement abstract vector spaces. It presumes we have "nice" standard bases available and can apply an intermediate coordinatization ourselves.

Exercise 1. In P_3, the vector space of polynomials with degree at most 3, find the vector representation of $p = \left(x^3 + x^2 + \frac{1}{2}\,x - \frac{33}{14}\right)$ relative to the basis for P_3:

$$B = \{5x^3 + 2x^2 + x + 1,\ -8x^3 - 3x^2 - x - 2,\\ 7x^3 + 4x^2 + x + 2,\ -7x^3 + 3x^2 + x - 2\}.$$

Hint: Coordinatize with respect to the basis $\left\{1, x, x^2, x^3\right\}$.

```
A = matrix(QQ, [[1, -2, 2, -2], [1, -1, 1, 1], [2, -3, 4,
    3], [5, -8, 7, -7]])
B = A.columns()
B
```

`B` is a basis, since `A` is nonsingular.

```
A.is_singular()
```

Now coordinatize `p`.

```
p = vector(QQ, [-33/14, 1/2, 1, 1])
p
```

We'll get a coordinatization old-style.

```
aug = column_matrix(B + [p])
r = aug.rref()
r
```

Let's check to see if this is right and we can recover `p`.

```
soln = r.column(4)
round_trip = sum([soln[i]*B[i] for i in range(4)])
round_trip, round_trip == p
```