Math 480a -- April 25, 2008 -- linear algebra

Summary of Wednesday's Student Feedback

1. Course is not too hard. Pacing is fine.
2. Want to know more about underlying algorithms.
3. Want more in depth homework assignments.
4. Several people said they couldn't find the pdf's and worksheets for lectures. This is mystifying since I've posted every single one before every lecture. The pdf input is sometimes truncated though before sage-3.0.
5. Want more about how to be Sage developer, e.g., coding conventions, navigating the Sage source code, coding environments, getting code into Sage, referee process?
6. Projects: Some worry. By the way, I keep thinking -- maybe projects are something you can plan to have done possibly weeks before class ends!?

Open Source Software quote of the day (from my inbox)

```
Timothy G Abbott
to      William Stein ,
date    Fri, Apr 25, 2008 at 12:02 AM

And thanks for making Sage such a great community to be a part of -- 
sage-devel mailing list is more friendly and positive than that in an
other free software community I've participated in (the comparison is
particularly jarring with Debian, which as you may know, is famous fo
its flamewars).

Debian is famous for its flamewars on their public mailing lists (occ
they even get news coverage).

I think it's really an issue of organizational structure -- the proje
very democratic and has a weak annually elected leader (and they drov
founder away from the project), and so despite the fact many Debian d
want the flamewars to stop, nobody has the authority to actually make
people stop (there are definitely people who do try).
```

```
        -Tim Abbott   [MIT CS grad student; Debian-Athena project dire
```

Linear Algebra: Vector Spaces and Modules

This lecture is about how to define vector spaces and modules in Sage, and do various calculations with them. In some cases we will explain some of the underlying algorithms used by Sage.

Vector Spaces

Recall that a **field** K is a set with a multiplication and addition that satisfies certain axioms; in particular, every nonzero element is invertible. Also, an additive abelian group is a set with an addition/subtraction operation that satisfies certain axiom; in particular $a + b = b + a$.

Definition: Vector Space A *vector space* V over a field K is an additive abelian group V equipped with a way of multiplying the elements of V by elements of K. More precisely, for $a, b \in K$ and $v, w \in V$ we have in addition to the group axioms for V and the field axioms for K that

$$a(v+w) = av + aw,$$

$$(a+b)v = av + bv,$$

$$(ab)v = a(bv),$$

and

$$1v = v.$$

```
%hide
A = sum([arrow3d((0,0,0), (random()-.5, random()-.5, random()-.5),
radius=0.01,color=hue(random()))
           for _ in range(30)])
A.show(aspect_ratio=[1,1,1], frame=False, spin=True)
```

Creating Vector Spaces

```
help(VectorSpace)
```

```
    Help on function VectorSpace in module sage.modules.free_module:

    VectorSpace(K, dimension, sparse=False, inner_product_matrix=None)
        EXAMPLES:
        The base can be complicated, as long as it is a field.
            sage: V = VectorSpace(FractionField(PolynomialRing(ZZ,'x')),3)
            sage: V
            Vector space of dimension 3 over Fraction Field of
    Univariate Polynomial Ring in x over Integer Ring
            sage: V.basis()
            [
            (1, 0, 0),
            (0, 1, 0),
            (0, 0, 1)
            ]

        The base must be a field or a \code{TypeError} is raised.
            sage: VectorSpace(ZZ,5)
    Traceback (click to the left for traceback)
    ...
            TypeError: K must be a field
```

```
V = VectorSpace(QQ, 2)
V
```

```
    Vector space of dimension 2 over Rational Field
```

```
type(V)
```

```
    <class 'sage.modules.free_module.FreeModule_ambient_field'>
```

```
category(V)
```

```
    Category of vector spaces over Rational Field
```

```
w = V([1,3/4])
```

```
w
```

```
    (1, 3/4)
```

```
type(w)
```

```
    <type 'sage.modules.vector_rational_dense.Vector_rational_dense'>
```

```
parent(w)
```

```
    Vector space of dimension 2 over Rational Field
```

```
10*w
```

```
    (10, 15/2)
```

```
VectorSpace(GF(17), 10000)
```

```
    Vector space of dimension 10000 over Finite Field of size 17
```

```
VectorSpace(CDF, 4)
```

```
    Vector space of dimension 4 over Complex Double Field
```

Ambient vector spaces are just tuples spaces; they can also be made with the following very nice shorthand

```
QQ^2
```

```
    Vector space of dimension 2 over Rational Field
```

```
CC^3
```

```
    Vector space of dimension 3 over Complex Field with 53 bits of
    precision
```

```
(CC^1000).dimension()
```

```
    1000
```

Basically we're doing **nothing interesting** above!

Subspaces

Given a vector space V one can create a subspace, which is a subset of a vector space that is also a vector space itself. Use the span command.

```
V = QQ^3; V
```

```
    Vector space of dimension 3 over Rational Field
```

```
W = V.span([[1,3,4], [-1,5,2], [0,8,6]])
```

```
W
```

```
    Vector space of degree 3 and dimension 2 over Rational Field
    Basis matrix:
    [  1   0 7/4]
    [  0   1 3/4]
```

```
vector(QQ, [1,3,4]) in W
```

```
    True
```

```
vector(QQ, [3,-2,15]) in W
```

```
    False
```

```
var('x,y')
plot3d(7/4*x + 3/4*y, (x,-5,5), (y, -5,5)) + \
   arrow3d((0,0,0), (1,3,4), color='red', radius=.1) +
arrow3d((0,0,0), (3,-2,15), color='purple', radius=.1)
```

```

```

Note above that the result is stored with a basis matrix in reduced row echelon form.

You can choose your own basis

```
W = V.span_of_basis([[1,3,4], [-1,5,2]])
W
```

```
    Vector space of degree 3 and dimension 2 over Rational Field
    User basis matrix:
    [ 1  3  4]
    [-1  5  2]
```

Subspaces also arise as kernels and images of matrices

```
a = random_matrix(QQ, 3, 5)
```

```
V = a.right_kernel()
V
```

```
    Vector space of degree 5 and dimension 2 over Rational Field
    Basis matrix:
    [1 0 0 0 1]
    [0 1 1 2 4]
```

```
a * V.0
```

```
    (0, 0, 0)
```

```
a * V.1
```

```
    (0, 0, 0)
```

Intersections and sums of subspaces

```
V = QQ^3
V1 = V.span([[1,3,4], [-1,5,2]])
V2 = V.span([[1,5,-1], [0,2,1]])
```

```
V1
```

```
    Vector space of degree 3 and dimension 2 over Rational Field
    Basis matrix:
    [  1    0 7/4]
    [  0    1 3/4]
```

```
V2
```

```
    Vector space of degree 3 and dimension 2 over Rational Field
    Basis matrix:
    [   1    0 -7/2]
    [   0    1  1/2]
```

```
var('x,y')
P = plot3d(7/4*x + 3/4*y, (x,-5,5), (y, -5,5), opacity=.7,
color='purple') + \
   plot3d(-7/2*x + 1/2*y, (x,-5,5), (y,-5,5), color='red',
opacity=.7)
P
```

```
V1.intersection(V2)
```

```
    Vector space of degree 3 and dimension 1 over Rational Field
    Basis matrix:
    [  1 -21 -14]
```

```
var('s')
P + parametric_plot3d( (s, -21*s, -14*s), (s, -0.2,.2),
color='green', thickness=5)
```

```
V1 = random_matrix(QQ,8,15).row_space()
V2 = random_matrix(QQ,9,15).row_space()
```

What is the dimension of the output likely to be?

```
I = V1.intersection(V2)
I.dimension()
```

```
    2
```

```
I
```

```
    Vector space of degree 15 and dimension 2 over Rational Field
    Basis matrix:
    [                       1                       0
    -2527836354/4149448765  10266929516/4149448765
    -3056788267/1185556790     488084782/592778395
    -3103758733/4149448765 -14644155583/8298897530
    10062919644/4149448765        -4890882/6858593
    1567189803/1659779506   -187986537/1659779506
    1549491079/592778395    1532351783/829889753
    8857377164/4149448765]
    [                       0                       1
    -14626423439/4149448765  47136674346/4149448765
    -5509361751/592778395    2286723782/592778395
    -15750279668/4149448765 -23841512309/4149448765
    39032150974/4149448765        -38349746/6858593
    3027051548/829889753   -1031660101/829889753    3996447924/5927783
    7097461321/829889753  39140446809/4149448765]
```

How do we intersect V_1 and V_2? One can compute kernels of matrices using Echelon form (or a p-adic lifting algorithm):

1. Compute the left kernel K of the matrix whose rows form a basis for V_1. Let A be a matrix whose rows are a basis for K. Then the kernel of K is V_1.
2. Compute the kernel of A restricted to V_2. This equals $V_1 \cap V_2$.

```
V1 + V2
```

```
    Vector space of degree 15 and dimension 15 over Rational Field
    Basis matrix:
    [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
    [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
    [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
    [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
    [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
    [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
    [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
    [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
    [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
    [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
    [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
    [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]
    [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
    [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
    [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
```

Quotient spaces

```
V = QQ^15
V1 = random_matrix(QQ,8,15).row_space()
```

```
Q = V/V1
```

```
Q(V.0)
    (1, 0, 0, 0, 0, 0, 0)
```

```
Q.3
    (0, 0, 0, 1, 0, 0, 0)
```

```
Q.lift_map()(Q.3)
    (0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
```

Linear Transformations

```
V2 = QQ^2
V3 = QQ^3
```

```
phi = V.hom([V3([1,2,3]), V3([-3,0,4])])
```

```
phi
    Free module morphism defined by the matrix
    [ 1  2  3]
    [-3  0  4]
    Domain: Vector space of dimension 2 over Rational Field
    Codomain: Vector space of dimension 3 over Rational Field
```

```
phi.kernel()
```

```
    Vector space of degree 2 and dimension 0 over Rational Field
    Basis matrix:
    []
```

```
phi((2,3))
```

```
    (-7, 4, 18)
```

```
phi.codomain()
```

```
    Vector space of dimension 3 over Rational Field
```

```
phi.image()
```

```
    Vector space of degree 3 and dimension 2 over Rational Field
    Basis matrix:
    [   1    0 -4/3]
    [   0    1 13/6]
```

```
FreeModule(ZZ, 3)
```

```
    Ambient free module of rank 3 over the principal ideal domain
    Integer Ring
```

```
ZZ^10000
```

```
    Ambient free module of rank 10000 over the principal ideal domain
    Integer Ring
```

Modules

Recall that a **ring** R is like a field but without the requirement that every nonzero element is invertible.

Definition: Module A *module* M over a ring R is exactly like a vector space, except the field K is replaced by the ring R. That's the only difference in the definition. However, vector spaces are often much ``easier to use'' than modules.

Submodules

```
(ZZ^3).span([[1,2,3], [-1,2,5], [0,4,8]])
```

```
    Free module of degree 3 and rank 2 over Integer Ring
    Echelon basis matrix:
    [1 2 3]
    [0 4 8]
```

```
(ZZ^3).span([[1/5,2/5,3/5], [-1,2,5], [0,4,8]])
```

```
    Free module of degree 3 and rank 2 over Integer Ring
    Echelon basis matrix:
    [1/5 2/5 3/5]
    [  0   4   8]
```

Quotient Modules

NOT IMPLEMENTED! Some grad students -- e.g., Ursula Whitcher -- would really like this for their research.

```
V = ZZ^3
W = V.span([[1,2,3], [-1,2,5], [0,4,8]])
```

```
V/W
```

```
    Traceback (click to the left for traceback)
    ...
    TypeError: unsupported operand type(s) for /:
    'FreeModule_ambient_pid' and 'FreeModule_submodule_pid'
```