Lecture 3 vectors and matrices

Open Lecture 3 on class website: www.math.hawaii.edu/190
See Chapter 2 of text for detailed explanations.

Vectors are sequences of numbers.

Row Vectors

Enter the following in SciLab:

```
[1,2,3]          notation for row vectors
[8]= =8
a=[2 3 4]        separate entries with spaces or commas
b=[10,10,10]     commas preferred, required in Fortran
a+b, b-a         add, subtract the respective coordinates
2*a, a+1         scalar product and addition
a^2, 2^a
a*b              wrong dimensions for matrix multiplication
a.*b             pointwise multiplication is not matrix muliplication
```

`a(1),a(2),a(3),` $a(i) = i^{th}$ entry of vector a

`a($)`= last, `a($-1)`= next to last element.

Column Vectors

```
a=[4;3;2]        notation for column vectors
b
b'               transpose
b                why didn't b change?
b=b', b          b doesn't change unless a change is assigned.
a+b, b-a, a.*b   operations are performed component wise.
```

Vector operations

```
sum(a)       sum of entries ,              sum( [4;3;2] ) = 9
prod(a)      product of entries,           prod( [4;3;2] ) = 24
max(a)       largest entry                 max( [4;3;2] ) = 4
min(a)       smallest entry                min( [4;3;2] ) = 2
length(a)    number of entries in the vector, length( [4;3;2] )=3
```

(not the geometric length or magnitude of the vector)

`a(length(a))` = last element = `a($)`, `a(1)` = first.

Dot (inner) product

classwork problem 2.

$$[a_1, a_2, ..., a_n] \cdot [b_1, b_2, ..., b_n] = a_1 b_1 + a_2 b_2 + ... + a_n b_n$$

= the *dot product*. The `dot_product` function (built into Matlab but not Scilab) must work for vectors of any length without using "...".

Example e3.1 dot_product Write a function `dot_product(a,b)` for the dot product of vectors `a`, `b`

```
function P = dot_product(a,b)
  P = sum(a.*b)
endfunction
clc; disp(dot_product([1,2],[3,4]))//Answer 11
disp(dot_product([1,2,3],[4,5,6])) //Answer 32
```

Example e3.2 Write a function `H(a)` for the sum of the cubes of the components of a vector `a`, i.e., $a_1^3 + a_2{}^3 + ... + a_n{}^3$. Test it on user input. Must work on vectors of any length.

```
function x=H(a)
  x=sum(a^3)
endfunction
clc
a=input('Enter a vector. > ')     //try [1,2]
disp(H(a))    //on [1,2] ans. is 9
```

MAGNITUDE (GEOMETRIC LENGTH) $\sqrt{a_1^2 + a_2^2 + a_3^2 + \ldots + a_n^2}$

This formula only for vectors of length 3. The classwork function code below must work for vectors of any length.

CLASSWORK c3.1(2) mag Together.

Define a function `mag(a)` which calculates the geometric length of a. Entering the formula one step-and-test at a time, starting with fixed input. After the function, include lines to get the vector a from the user and to display `mag(a)`. If you enter "[1,2,3]" you should get 3.7416574

Copy lines, File/New in SciNotes, paste, File/Save as c3.1(2)mag(a)

```
//c3.1(2)mag    mag(a)=geometric length of a.
//Test on user input: a = input('Enter a vector. > ')
// Entering [1,2,3] should give 3.7416574
   //delete this line, write in the three lines for the function

clc;printf('\n')
a=input('Enter a vector.>')
disp(mag(a))
```

MATRICES

Enter in SciLab: -- repeat all these steps at home.

```
a=[1 2; 3 5]
b=[1,1;1,1]
```

`[a,b]` a followed on the right by b.

`[a;b]` a, with b appended below it.

```
2*b,a+b, a-b, a.*b, a*b
```

`[n,m]=size(a)` // n= number of rows, m= number columns

`a(1,2),a(2,1)` a(i,j) = entry in i^{th} row, j^{th} column

`a(2,2)=4` changes a(2,2) to 4

```
a,a(:,:)
```

`a,a(:,1),a(:,2)` a(:, j)= all rows of j^{th} column= j^{th} column

`a,a(1,:),a(2,:)` a(i, :)= all columns of i^{th} row = i^{th} row

```
a,
a(1,2)=10
a(:,2)=9
a(1,:)=[7,8]
b
```

`z=b` saves b to z

`b(1,:)= b(1,:)+2` What does this do?

`b=z` recovers b from z

`b(:,1)= b(:,1)-2` What does this do?

`a([2,1],:)` result of swapping rows 1, 2 //homework problem

`a(:,[2,1])` result of swapping columns 1,2

```
zeros(2,3)
ones(2,3)
8*ones(2,3)
```

MATRIX MULTIPLICATION, IDENTITY MATRIX, MATRIX INVERSES

`a*b` matrix multiplication

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{21} + a_{22}b_{22} \end{bmatrix}$$

The entry a_{ij} in the i^{th} row and j^{th} column of the product = dot (or inner) product of the i^{th} row of the first matrix and the j^{th} column of the second matrix.

`I=eye(2,2)` This is Scilab notation, Matlab uses `eye(2)`.

I is the 2×2 identity matrix.

1 is the identity for multiplication of numbers: $1x = x1 = x$.

I is the identity for matrix multiplication: $I * a = a * I = a$

`a, I*a, a*I,`

x^{-1} is the inverse operation of multiplication of numbers:

$x(x^{-1}) = (x^{-1})x = 1$.

x^{-1} is also the inverse operation of multiplication of matrices:

$a * a^{-1} = a^{-1} * a = I$

In Matlab/Scilab, the inverse a^{-1} is `inv(a).`

```
a=[1,2;3,4]
inv(a)
a*inv(a),inv(a)*a
```

`2.220D-16` $= 2.22 \times 10^{-16} \approx 0$

Computer arithmetic isn't always exact.

1/3 = .3333333333333333....

$\approx$ .333333

Classwork c3.2(3) id

Together. Write a function `id(n)` which generates the `n`×`n` identity matrix. Test on `n=4,6`.

Copy lines, File/New in SciNotes, paste, File/Save as c3.2(3)id

```
//c3.2(3)id    id(n)=nxn identity matrix.
        //delete this line, fill in the function

clc; disp(id(3)); disp(id(4))
```

Classwork c3.3(3) add_mult

Together. Write a function `add_mult(a,i,r,j)` which adds `r` times row `j` to row `i`. Test `add_mult(a,2,8,3)` for `a=[1,0,0;01,0;0,0,1]`.

Copy lines, File/New in SciNotes, paste, File/Save as c3.3(3)add_mult

```
//c3.3(3)add_mult   add_mult(a,i,r,j)
        //delete this line, fill in the function

a=[1,0,0;0,1,0;0,0,1]; clc
disp(a); disp(add_mult(a,2,8,3))
```

Classwork problems due end of class

c3.1(2)mag, c3.2(3)id, c3.3(3)add_mult

email to: `dale@math.hawaii.edu` subject line: `190 c3(8)`

To enable automatic routing, copy the subject line exactly.

Closed-book quiz at beginning of class

No computer; no text. Like the Hw problems. Be on time. There will be one problem. It will be like a homework problem.

Homework h3.1(3) distance.

The distance between points

$a = [a_1, a_2, ..., a_n]$ and $b = [b_1, b_2, ..., b_n]$ is

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}$$

Write a Scilab function for the distance `D=distance(a,b)` between vectors `a` and `b`. Use it to find the distance between `[1,2]` and `[3,3]` and the distance between `[1,2,3]` and `[3,3,3]`. Your function must work for vectors of any length, it may not use +...+. See example **E3.2** above. Try entering the formula one step-and-test at a time.

```
//h3.1(3)distance  Use it to find the
//distance between [1,2],[3,3] Answer: 2.23
//distance between [1,2,3],[3,3,3] Answer: 2.23
```

Homework h3.2(3) swap

Write a Scilab function `swap(a,i,j)` which swaps rows `i` and `j`. For example,

$$\text{swap}\left(\begin{matrix}1&2&3\\4&5&6\\7&8&9\end{matrix},2,3\right)=\begin{matrix}1&2&3\\7&8&9\\4&5&6\end{matrix}$$

Fill in the blank line. See example above.

```
//h3.2(3)swap  swaps rows i, j
function b=swap(a,i,j)
    b=a
    b([i,j],:)= _______________
endfunction
//Testing lines
a=[1,2,3;4,5,6;7,8,9]; disp(a),disp(swap(a,1,3))
a=[3 3; 6 6]; disp(a),disp(swap(a,1,2))
```

Homework h3.3(3) mult

Write a Scilab function `mult(a,i,r)`which multiplies row `i` by r. For example,

$$\text{mult}\left(\begin{matrix}1&2&3\\4&5&6\\7&8&9\end{matrix},2,10\right)=\begin{matrix}1&2&3\\40&50&60\\7&8&9\end{matrix}$$

```
//h3.3(3)mult  mult(a,i,r) multiplies row i by r
//test on a=[1,0,0;0,1,0;0,0,1], mult(a,2,8)
```

Homework due before next class

Write these from scratch.

h3.1(3)distance, h3.2(3)swap, h3.3(3)mult

email to: `dale@math.hawaii.edu` subject line: `190 h3(9)`

Elementary row operations

Given a matrix, there are three types of elementary row operations: You may

- switch (permute) rows, -- `swap`
- multiply a row by a nonzero constant, -- `mult`
- add a multiple of one row to another row -- `add_mult`

To pivot on the *i*-*j*th entry of a matrix (assumed nonzero) means using elementary row operations to make that entry 1 and all other entries in the *j*th column 0. The leading coefficient of a row is the first nonzero entry. A matrix is in refuced row echelon form (rref) iff the leading coefficient of each row is 1 and it is to the right of the leading coefficient of the previous row.

Classwork c4.1(3) pivot

Write a function `pivot(a,i,j)` which pivots matrix `a` on row `i` and column `j`. Assume `a(i,j)`$\neq$ `0`. Use a sequence of these pivots and swaps (see h3.2(3)) which converts `a`=[0,2,3,5;2,3,4,6;5,6,6,7] to reduced row echelon form (rref)

Copy lines, File/New in SciNotes, paste, File/Save as c4.1(3)pivot

```
//c4.1(3)pivot   pivot(a,i,j) pivots on a(i,j)
//copy the code for swap here (not its testing lines).
        //delete this line, fill in the function
a=[0,2,3,5;2,3,4,6;5,6,6,7]
//convert a to rref.
clc;disp(a)
        //delete this line, add a sequence of pivots and swaps to get rref
```

Homework h4.1(3) pivot

Use the classwork function `pivot(a,i,j)`to convert `a`=[3,0,3,5;2,0,4,6;5,6,6,7] to reduced row echelon form.

```
//h4.1(3)pivot   add lines to convert
//a=[3,0,3,5;2,0,4,6;5,6,6,7] to rref.
```