Scilab Quick Ref-Card

Basic operators

`+`	addition for scalers, vectors, matrices etc.
`-`	subtraction for scalers, vectors, matrices etc.
`\`	$2\backslash 7 = 3.5$ i.e. 2 divides 7 $\frac{7}{2}$
`/`	$2/7 = 0.2857143$ i.e. fraction $\frac{2}{7}$
`*`	multiplication of scaler, matrices
`^`	2^3 computes 2^3
`//`	comment line
`:`	Defines range `1:5` will print sequence of numbers as 1. 2. 3. 4. 5. with default increment as 1
`;`	to suppress output

Data types / objects

scaler	`x=3` Assigns value to variable x which becomes scaler.
range	`2 : .5 : 4` will print sequence of numbers starting from 2 with of increment .5 in the range 2 to 4.
`linspace`	e.g. `linspace(2,3,11)` will return sequence of equally space points in the interval $[2,3]$
vector	`U=[2,9,-4]` U becomes the vector of length 3
matrix	`A=[3,4;5,11]` $A = \begin{bmatrix} 3 & 4 \\ 5 & 11 \end{bmatrix}$; separates rows.
complex number	`3+%i` This will dislay complex number $3+i$
variable x	$x = poly(0,'x')$ This will declare x as a variable x which is soln of polynomial$x = 0$.

Predefined constants

`%pi`	π with numerical value assigned
`%e`	e base in natural logarithm $e = 2.7182818$
`%i`	complex number $\sqrt{-1}$
`%inf`	Infinity
`%eps`	machine epsilon

Extracting elements, rows & columns

`X(n)`	nth element of vector X
`X($)`	last element of vector X
`X($-1)`	second last element of vector X
`A(2,3)`	element of matrix A at 2nd row, 3rd column
`A(:,2)`	In matrix A second column, `:`all rows
`A(n,:)`	nth row and all entries in nth row
`A(:,$)`	Last column of matrix A
`A($-1,:)`	Second last row
`A(2:3,1:2)`	submatrix i.e. entries from 2nd to 3rd row and 1st to 2nd column of matrix A

Advanced operations

`.*`	element wise multiplication, applicable to vectors, matrices
`./`	U, V are vectors(or matrices), `U./V` is $u_1./v_1, \cdots, u_k./v_k$, in other words $\frac{u_1}{v_1}, \cdots, \frac{u_k}{v_k}$
`.\`	U, V are vectors, `U.\V` is $v_1.\backslash u_1, \cdots, v_k.\backslash u_k$, in other words $\frac{v_1}{u_1}, \cdots, \frac{v_k}{u_k}$
`A \ B`	Matrix A divides matrix B i.e. $A^{-1} \times B$
`A / B`	Matrix B divides matrix A i.e. $A \times B^{-1}$
`n!`	returns n factorial
`;`	2/3; will supress output

Logical operators

Logical operations returns answers as TRUE or FALSE

$a < b$	is a strictly less than b?
$0 > d$	is d less than 0?
$m <= n$	is m is less than or equal to n?
$a >= 0$	is a great than on equal to 0?
$A <> B$	is A not equal to B
$a == b$	is a equals to b?

Standard functions

Trigonometric functions

Accepts input in radians. For degrees use `sind(45)`

`sin`	sin
`cos`	cos
`tan`	tan
`sec`	sec
`csc`	cosec
`cotg`	cot
`asin`	sin inverse
`acos`	cos inverse
`atan`	tan inverse
`asec`	sec inverse
`acsc`	cosec inverse
`acot`	cot inverse

Mathematical functions

`exp`	exponential
`log`	logarithm
`round`	rounding
`floor`	earlier highest integer
`ceil`	next lowest integer
`int`	only integer part
`modulo(x,y)`	$x \pmod y$ gives remainder
`factorial(n)`	will out put $n!$
`factors(a)`	will factor number a in prime numbers
`sqrt(5)`	square root of a number
`abs(-5)`	absolute value of a number

Miscellaneous functions

`clear a`	delete the predefined object a
`disp(x)`	Prints value of x
`disp(‘‘abc’’)`	Prints string as it is
`find`	`find(A<3)` will return index value (positions of values) in A which are < 3
`clean(x)`	will round x to 0
`format('v',20)`	set number of digits to 20
`format('e',20)`	represent number in scientific format e.g. 2345=2.345D+03 which is equivalent to 2.345×10^3
`help("plot")`	display help for `plot` command

Matrix related functions

If A is some matrix already defined.

`length(A)`	It will return number of elements in A.
`size(A)`	return number of rows, columns in A
`sum(A)`	Addition of all elements of A
`prod(A)`	Product of all elements
`'`	transpose of matrix, e.g. `A'`, where A is a matrix
`trace(A)`	Addition of all diagonal elements
`diag(A)`	Extract all diagonal elements
`max(A)`	maximum element in matrix A
`min(A)`	minimum element in matrix A
`rank(A)`	rank of matrix A
`det(A)`	It will find the determinant of the square matrix.
`inv(A)`	if determinant is non singular (i.e. $\neq 0$) then it will compute the inverse of the matrix
`spec(A)`	spectrum- it will compute eigen values
`[V,E]=spec(A)`	will assign eigen values to E in diagonal matrix form will assign corresponding eigen vectors in column form to matrix V

Polynomial operations

`x=poly(0,'x')`	declare x as poly with variable x and root= 0
`x=poly(v,'x','coef')`	declare poly with variable x & coefficients from vector v
`roots(f)`	will find roots of polynomial f.
`polfact(f)`	will factorize polynomial f.
`derivat(f)`	derivative of polynomial f
`horner(f,5)`	evaluate poly f at 5.
`coeff(f)`	returns coefficient of the poly f.
`degree(f)`	returns degree of poly f

Special Matrices

`eye(3,3)`	Identity matrix of size 3×3
`ones(3,3)`	Matrix of size 3×3 with each element= '1'.
`zeros(1,2)`	Matrix of size 1×2 with each element = '0'.
`rand(2,3)`	Matrix of size 2×3 with entries generated randomly between 0 and 1
`diag([2 -5 7])`	will output matrix as $\begin{bmatrix} 2 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 7 \end{bmatrix}$

Programming

```
Inline function defination
deff ('y=funname(x)','y=2*sin(x)')
y is temporary variable.
funname is name of the function.
y=2*sin(x) is function definition
For loop

k=0
for i=1:n
  k=3*i+1
end

if (condn)  then ...end
if (condn) then...else...end
while(codn)...end
```

```
function[output]=funname(input)
output=calculation
endfunction
```

```
function[f]=fibbo(n)
  f=[ 1 1]
   temp=0
   for i=2:n
     temp=f($) +f($-1)
      f=[f  temp]
    end
endfunction
continue:  continues with next
counter value, skip the current counter
```

Graphics

`plot(sin(x))`	`plot sin graph Vs index value`
`plot(x,y)`	`graph of`x `Vs` y
`plot(x,y,x,w)`	
`plot2d`	
`fplot2d`	`2-d function plot`
`fplot3d`	`3-d function plot`
`fplot3d1`	`colourful 3D graph`
`subplot()`	`divide plot windows`
`xlabel()`	`horizontal label to graph`
`ylabel()`	`vertical label`
`legend()`	`list of graphs with colours used`
`clf()`	`close graphics window`
`pie()`	`pie graph`
`contour`	`contour plot`
`champ`	`vector field plot`
`champ1`	

Advanced Matrix related functions

`rref`	`Row reduced echelon form returns identity matrix and diagonal matrix`
$[A \;\; B]$	`Augmented matrix is possible by writing one matrix followed by the other one`
`svd`	`returns singular value decomposition`
`LU`	`LU decomposition returns upper triangular, lower triangular, upper triangular, & Identity matrix or pivoted matrix`
`bdiag`	`block matrix`

Points to be noted

- `function returns more than one value.`
- `append operation is possible with` $a = [a, 2]$.
- `Matrix in one variable (defined by poly command).`
- `works in same way.  Operations for matrix works with it.`
- `matrix, polynomial are also valid inputs for functions.`
- `Scilab acts on whole vector at a time, use of loops can be avoided.`

Manjusha Joshi   email:manjusha.joshi@gmail.com August 13, 2014