Linear Algebra Using SymPy Cheat Sheet

by royqh1979 via cheatography.com/87753/cs/20231/

Import SymPy

import sympy as sp

Matrix Creation

normal Matrix	sp.**Matrix**([[1,2],[3,4]])
Matrix with all zeros	sp.**zeros**(4,5)
Matrix with all ones	sp.**ones**(4,5)
Square matrix with all zeros	sp.**zeros**(5)
Square matrix with all ones	sp.**ones**(5)
Identity matrix	sp.**eyes**(5)
Diagonal Matrix	sp.**diag**(1,2,3,4)
Generate element with func(i,j)	sp.**Matrix**(2,3,func)

Matrix Modification

Delete the i-th row	M.row_del(i)
Delete the j-th column	M.col_del(j)
Row join M1 and M2	M1.row_join(M2)
Column join M1 and M2	M1.col_join(M2)

Indexing(Slicing)

get the element in M at (i,j)	M[i,j]
get the i-th row in M	M.row(i)
get the i-th row in M	M[i,:]
get the j-th column in M	M.col(j)
get the j-th column in M	M[:,j]
get the i-th and the k-th rows	M[[i,k],:]
get the j-th and the k-th columns	M[:,[j,k]]
get rows from i to k	M[i:k,:]
get columns from j to k	M[:,j:k]
get sub-matrix (row i to k,col j to l)	M[i:k,j:l]

Note: All indices start from 0

Basic opertaions

Sum	A+B
Substraction	A-B
Matrix Multiply	A*B
Scalar Multiply	5*A
Elementwise product	sp.matrix_multiply_elementwise(A,B)
Transpose	A.T
Determinant	A.det()
Inverse	A.inv()
Condition Number	A.condition_number()
Row count	A.rows
Column count	A.cols
Trace	A.trace()

Elementary Row Operations

Replacement	m.row_op(i, **lambda** ele,col:ele+m.row(j)[col]*c)
Interchange	M.row_swap(i,j)
Scaling	m.row_op(i,**lambda** ele,col:ele*c)

Linear Equations

Echelon From	M.echelon_form()
Reduced Echelon Form	M.rref()
Solve AX=B (B can be a matrix)	x,freevars=A.gauss_jordan_solve(B)
least-square fit Ax=b	A.solve_least_squares(b)
solve Ax=b	A.solve(b)

Vector Space

Basis of column space	M.columnspace()
Basis of null space	M.nullspace()
Basis of row space	M.rowspace()
Rank	M.rank()

Eigenvalues amd Eigenvectors

Find the eigenvalues	M.eigenvals()
Find the eignevalues and the corresponding eigenspace	M.eigenvects()
Diagonalize a matrix	P, D = M.diagonalize()
test if the matrix is diagonalizable	M.is_diagonalizable
Calculate Jordan From	P, J = M.jordan_form()

Decomposition

LU Decomposition(PA=LU)	P,L,U=A.LUdecomposition()
QR Decomposition	Q,R=A.QRdecomposition()

Vector Operations

Create a column vector	v=sp.Matrix([1,2,3])
dot product	v1.dot(v2)
cross product	v1.cross(v2)
length of the vector	v.norm()
normalize of vector	v.normalize()
the projection of v1 on v2	v1.project(v2)
Gram-Schmidt orthogonalize	sp.GramSchmidt([v1,v2,v3])
Gram-Schmidt orthogonalize with normalization	sp.GramSchmidt([v1,v2,v3],True)
Singular values	M.singlular_values()

Block Matrix

Create a matrix by block	M=sp.Matrix([[A,B],[C,D]])

Useful Links

Useful Links
SymPy Documentation
SymPy Tutorial