

### Libraries

Numpy	<code>import numpy as np</code>
Linear algebra (numpy)	<code>import numpy.linalg as LA</code>
Sparse operations (scipy)	<code>import scipy.sparse as ss</code>
Sparse normalizations (sklearn)	<code>from sklearn.preprocessing import normalize</code> <code>_norm</code>

### Array comparisons

Locate indices based on condition	<code>index = np.where(array == 4)</code>
Check if all/any satisfy condition	<code>np.any(array == 1), np.all(array1 == array2)</code>

### Matricial operations

Dot product	<code>np.dot(v1, v2)</code> or <code>v1 @ v2</code>
Outer product	<code>np.outer(v1, v1)</code>
Sum rows (0), columns (1)	<code>np.sum(A, axis=0)</code>
Eigenvector, eigenvalues	<code>eigval, eigvec = LA.eig(A)</code>

The eigenvector output of the `eig()` function is a matrix containing the usual eigenvectors as its columns.

### Numpy comparisons

Check if any element satisfies a condition	<code>np.any(A == 1)</code>
Find which elements (their indices) satisfy a condition	<code>np.where(A == 1)</code>
Check if all elements satisfies a condition	<code>np.all(A == 1)</code>
Closeness up to tolerance	<code>np.isclose(1.001, 1)</code>

### Numpy utilities

Arrange all array elements in a list	<code>np.ravel(A)</code>
--------------------------------------	--------------------------

### Sparse matrices with scipy

Convert numpy matrices to sparse	<code>A_sparse = ss.csr_matrix(A)</code> <code>A_sparse = ss.csc_matrix(A)</code>
Change CSC to CSR and viceversa	<code>A_sparse_csr = A_csc.tocsr()</code> <code>A_csc = A_csr.tocsc()</code>
Row/col. normalization	<code>A_norm = sparse_normalize(A_csc, norm=1)</code>
Sparse identity matrix	<code>sparse_id = ss.identity(n)</code>
Invert a sparse matrix (CSC)	<code>sparse_inv = ss.inv(A_sparse)</code>
Dot product with vector	<code>A_sparse.dot(v)</code>
Transpose sparse matrix	<code>At_sparse = A_sparse.transpose()</code>

CSC (*Compressed Sparse Column*) are more suitable for column operations while CSR (*Compressed Sparse Row*) are for row operations. For instance, row/col. normalizations.

