# ACSE_Ia

**Orkan Sezer**

**Oct 29, 2020**

# CONTENTS

CHAPTER

# ONE

# A GAUSSIAN ELIMINATION ROUTINE

This package implements Gaussian elimination[1] for `numpy.ndarray` objects, along with hand-written matrix multiplication.

See *acse_la.gauss()*, *acse_la.matmul()* and *acse_la.zeromat()* for more information.

`acse_la.`**`gauss`**(*a*, *b*)

Given two matrices, *a* and *b*, with *a* square, the determinant of *a* and a matrix *x* such that a*x = b are returned. If *b* is the identity, then *x* is the inverse of *a*.

**Parameters**

- **a** (*np.array or list of lists*) – 'n x n' array
- **b** (*np. array or list of lists*) – 'm x n' array

### Examples

>> a = [[2, 0, -1], [0, 5, 6], [0, -1, 1]] >> b = [[2], [1], [2]] >> det, x = gauss(a, b) >> det 22.0 >> x [[1.5], [-1.0], [1.0]] >> A = [[1, 0, -1], [-2, 3, 0], [1, -3, 2]] >> I = [[1, 0, 0], [0, 1, 0], [0, 0, 1]] >> Det, Ainv = gauss(A, I) >> Det 3.0 >> Ainv [[2.0, 1.0, 1.0], [1.3333333333333333, 1.0, 0.6666666666666666], [1.0, 1.0, 1.0]]

### Notes

See https://en.wikipedia.org/wiki/Gaussian_elimination for further details.

`acse_la.`**`matmul`**(*a*, *b*)

Given two matrices, *a* and *b*, first checks if the the dimensions of 'a' and 'b' are compatible for multiplication. From matrix algebra, we know that for a*b to exist, and if a is an n x p matrix and b is a p1 x q matrix, then p = p1 must hold true. The resultant matrix c (c = a*b) which is an n x q matrix is then created as a zeros matrix and corresponding matrix elements are stored via traditional matrix multiplication, i.e. the dot product of the i_th row of a and the j_th column of b are stored as c[i][j].

**Parameters**

- **a** (*np.array or list of lists*) – 'n x p' array
- **b** (*np. array or list of lists*) – 'p1 x q' array

[1] https://mathworld.wolfram.com/GaussianElimination.html

### Examples

>> a = [[1, 2, 3], [4, 5, 6]] >> b = [[10, 11], [20, 21], [30, 31]] >> c = matmul(a, b) >> c [[140, 146], [320, 335]] >> A = [[1, 0, -1]] >> B = [[1, 0, 0], [1, 1, 0], [6, 4, 1]] >> C = matmul(A, B) >> C ValueError: Incompatible dimensions

acse_la.**zeromat**(*p*, *q*)

Creates a p x q zero matrix, meaning that the new matrix has 0 for all its entries.

**Parameters**

- **p** (*integer*) – number of rows of zeros matrix
- **q** (*integer*) – number of columns of zeros matrix

### Examples

>> p = 3 >> q = 4 >> z_mat = zeromat(p, q) >> z_mat [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]

## References

# PYTHON MODULE INDEX

## a

# INDEX