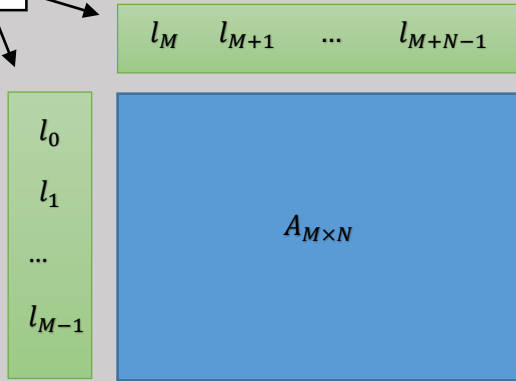
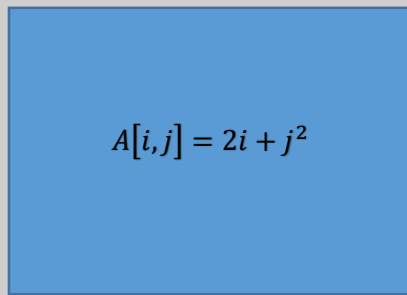


labels



```
// example for M = 6, N = 3
size_t dim_count = 2;
size_t dims[] = {6,3};
struct matrix *A = matrix_init(dim_count, dims);
printf("A: \n");
matrix_print_metadata(A);
matrix_print(A);
// or just use matrix_print_all(A) instead of
previous 3 lines
matrix_destroy(A);
```



```
size_t pos[2];
int i, j;
for (i=1; i<=6; i++) {
    for (j=1; j<=3; j++) {
        pos[0] = i;
        pos[1] = j;
        matrix_set_element(A, pos, 2*i+j*j);
    }
}
```

$t_{1 \times 10} = \text{linspace}(0, \pi, 10)$

$A_{1 \times 10}[i] = \sin(t[i])$

```
struct matrix *t = matrix_linspace(0, MATRIX_PI, 10);
struct matrix *A = matrix_sin(t);
matrix_destroy_batch(2, t, A);
```

$x_{1 \times 10} = \text{linspace}(0, \pi, 10)$

$A_{7 \times 10}(x, y) = \sin(x) * \cos(y)$
 $\equiv A[i, j] = \sin(x[i]) * \cos(y[j])$

```
struct matrix *x = matrix_sin_linspace(0, MATRIX_PI, 10);
struct matrix *y = matrix_cos_range(0, 30, 5);
struct matrix *A = matrix_crossproduct(y, x);
matrix_print_all(A);
matrix_destroy_batch(3, x, y, A);
```

$y_{1 \times 7} = 0:5:30$

$l_M, l_{N+1}, \dots, l_{N+M-1}$

$A_{M \times N}^T = A_{N \times M}^T$

```
// struct matrix *y = matrix_cos_range(0, 30, 5) to a 6x3
matrix
size_t new_dim_count = 2;
size_t new_dims[] = {6,3}
struct matrix *y1 = matrix_reshape(A, new_dim_count,
new_dims);
```

```
struct matrix *AT = matrix_transpose(A);
```