Table of Contents

1. dotprod

```
dotprod = @(x, y) (x)' * y;
```

2

```
function [orthonormal] = is_orthonormal(array, func)
    num_cols = size(array, 2);
    tolerance = 1000 * eps;
    for k = 1:num_cols
        if abs(sqrt(func(array(:,k), array(:,k))) - 1) > tolerance
            orthonormal = false;
            return
        end
        for j = k+1:num_cols
            if abs(func(array(:,k), array(:,j))) > tolerance
                orthonormal = false;
                return
            end
        end
    end
    orthonormal = true;
    return
end
```

```
gram_schmidt_test_matrix =

   0.3737 + 0.1121i  -0.2958 + 0.1156i  -0.4423 + 0.4800i  -0.4730 - 0.3117i
   0.3363 + 0.4858i  -0.3361 - 0.0837i   0.2496 - 0.4500i  -0.3618 + 0.3676i
   0.4111 + 0.3737i   0.7950 + 0.1875i   0.0469 - 0.0137i   0.0016 - 0.1475i
   0.3737 + 0.2242i  -0.2966 - 0.1551i   0.3775 + 0.4053i   0.6180 - 0.0967i


output =

  logical

   1
```

3 gram-schmidt

```
function [orthonormalArray] = gram_schmidt(array, func)
    [num_rows, num_cols] = size(array);
    if (is_orthonormal(array, func))
        orthonormalArray = array;
        return
    end
    orthonormalArray = zeros(num_rows, num_cols);
    for k = 1:num_cols
        v_k = array(:, k);
        for j = 1:k-1
            v_k = v_k - (func(orthonormalArray(:, j), v_k) *
orthonormalArray(:, j));
        end
        orthonormalArray(:, k) = v_k / sqrt(func(v_k, v_k));
    end



end
```

4 tester

```
test_matrix = randi(15,4,4) + 1j * randi(15, 4 ,4);
gram_schmidt_test_matrix = gram_schmidt(test_matrix, dotprod)
output = is_orthonormal(gram_schmidt_test_matrix, dotprod)
```

Published with MATLAB® R2025b