Contents


```
% SPDX-License-Identifier: GPL-3.0-or-later
%
% ECE210 assignment04.m -- Func Off
% Copyright (C) 2024 Aidan Cusa <aidancusa@gmail.com>

clc;    % clear command window
clear;  % clear all variables from current worwkspace
close all;
```

part 1

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

% inner product norm for L2
ip_norm = @(x) sqrt(ip(x, x));
```

part 4

```
S = [1, 2 + 3j, -1 + 7j
     1j, 3j, 6 + 10j
     2 - 1j, 1 - 1j, 11 - 4j
    -1, 2j, 3 + 4j];

U = gram_schmidt(S, ip_norm)
```

part 5

```
orthogonal = isorthogonal(U(:, 1), U(:, 2), ip) & ...
    isorthogonal(U(:, 2), U(:, 3), ip) & ...
    isorthogonal(U(:, 1), U(:, 3), ip)
```

part 2

taken inspiration from https://web.mit.edu/18.06/www/Essays/gramschmidtmat.pdfc

```
function gs = gram_schmidt(V, ip_norm)
    [m, n] = size(V);

    Q = zeros(m, n);
    R = zeros(n, n);
```

```
    for j = 1:n
        v = V(:, j);

        for i = 1:j-1

            R(i, j) = Q(:, i)' * V(:, j);
            v = v - R(i, j) * Q(:, i);
        end

        R(j, j) = ip_norm(v);
        Q(:, j) = v / R(j, j);

        gs = Q;
    end
end
```

```
U =

   0.3536 + 0.0000i   0.2236 + 0.6708i  -0.5916 + 0.0845i
   0.0000 + 0.3536i   0.0000 + 0.4472i   0.5071 + 0.1690i
   0.7071 - 0.3536i  -0.2236 - 0.0000i   0.4226 + 0.0000i
  -0.3536 + 0.0000i   0.2236 + 0.4472i   0.4226 + 0.0000i
```

part 3

```
function io = isorthogonal(u, v, ip)
    io = ip(u, v) < eps;
end
```

```
orthogonal =

  logical

   1
```

Published with MATLAB® R2023b