```
% Tests for Matching Solutions method without noise with 2↙
iterations

close all;
clearvars;

k = 16; % Number of sensors
m = 4; % Size of observation vectors b
n = 20; % Size of unknown vector x
reliable_sensors_list = [6 8 10 12 14]; % Number of consistent↙
sensors
restriction_delta = 10^-10;
threshold = 10^-4;
MCexperiments = 1000;

for s_index = 1:length(reliable_sensors_list)

    s = reliable_sensors_list(s_index);

    fprintf('Considered %d reliable sensors. ', s);

    reliable_sensors = [ones(1, s) zeros(1, k-s)];

    parfor j=1:MCexperiments

        %preallocations
        bi = zeros(m, 1, k);

        % unknown vector is modeled as x0 ~ N(0, n^(-1/2)In)
        x0 = mvnrnd(zeros(1, n), n^(-1)*eye(n))';

        % Entries of matrix A are drawn independently from N(0, 1)
        Ai = randn(m, n, k);

        % generate sensor observation data b
        % consistent observations
        for i=1:s
            bi(:, :, i) = Ai(:, : ,i)*x0;
        end
```

```
        % unreliable sensors
        for i=s+1:k
            bi(:, : , i) = mvnrnd(zeros(1, m), eye(m))';
        end

        x_est_0 = randn(n, k) / sqrt(n);
        lambda_est_0 = mean(x_est_0, 2);

        % lambda_est = matching_solutions( Ai, bi, n, k,↙
restriction_delta, x0, L0);
        [x_est_1, lambda_est_1] = matching_solutions_miter( Ai, bi,↙
n, k, restriction_delta, x_est_0, lambda_est_0);
        [x_est_2, lambda_est_2] = matching_solutions_miter( Ai, bi,↙
n, k, restriction_delta, x_est_1, lambda_est_1);

        % check for reliable sensors
        method_reliable_sensors_iter1 = sensor_validation( Ai, bi,↙
lambda_est_1, threshold, k, s);
        method_reliable_sensors_iter2 = sensor_validation( Ai, bi,↙
lambda_est_2, threshold, k, s);

        list_results_iter1(j, s_index) = isequal(reliable_sensors,↙
method_reliable_sensors_iter1);
        list_results_iter2(j, s_index) = isequal(reliable_sensors,↙
method_reliable_sensors_iter2);
    end
end

results(1, :) = sum(list_results_iter1, 1);
results(2, :) = sum(list_results_iter2, 1);

results = (results/MCexperiments) * 100;

% Print results
f = figure('Position',[440 500 500 140]);
% Create the column and row names in cell arrays
cnames = {'s=6','s=8','s=10', 's=12', 's=14'};
rnames = {'MS(1)', 'MS(2)'};
```

```
% Create the uitable
t = uitable(f,'Data',results,...
            'ColumnName',cnames,...
            'RowName',rnames);

% Set width and height
t.Position(3) = t.Extent(3);
t.Position(4) = t.Extent(4);
```