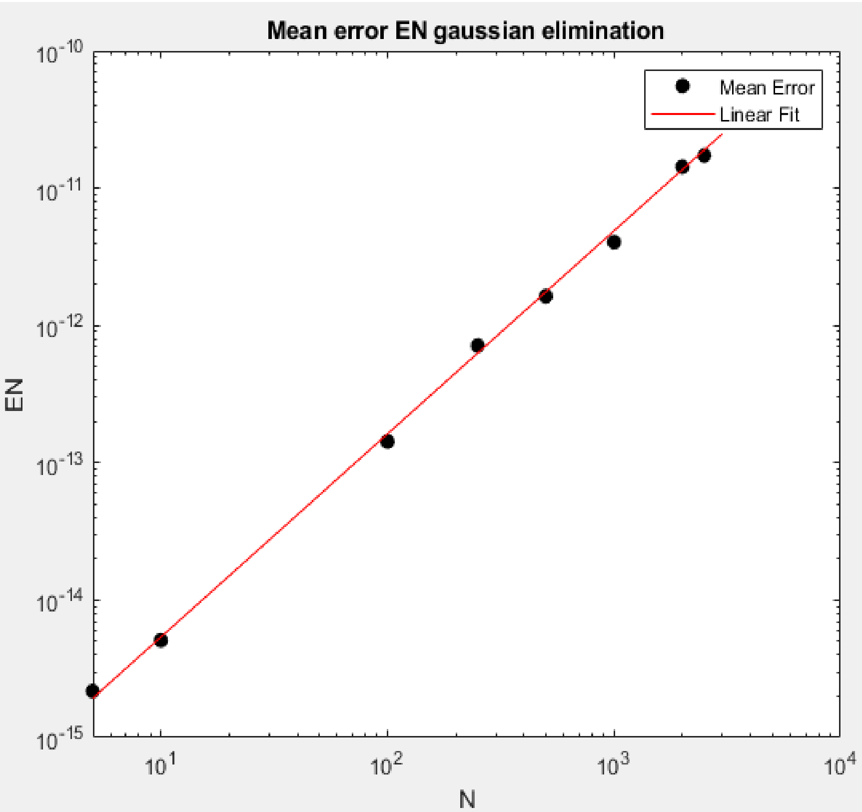
MACM 316: CA3 **AUTHOR: Andy Liu** **STUDENT ID: 301472847**



a) Function code is displayed on the next page.

b) For my choices of M and N, I chose M = 500 because it struck a good balance between performance and accuracy where code would take 5-10 min to run. I chose my N to range from N = 5 to N = 3000. Using this formula given in lecture slides:

- **Total operation count:**

$$\mathcal{T}(\mathbf{n}) = \mathcal{R}(\mathbf{n}) + \mathcal{B}(\mathbf{n}) = \frac{\mathbf{n}^3 + 3\mathbf{n}^2 - \mathbf{n}}{3}$$

I calculated the total operation count to range from 65 to around 9 billion which accurately tests behaviours of the algorithm with both a small number of operations and a large number. These N and M values were also the best that my computer could handle within 5-10 min.

c) Graph on the left plots the linear fit line.

d) The linear fit line accurately approximates the data as seen in the left figure.

e)

$$p_1 = 1.4291, \qquad p_2 = -15.6194$$

$$log_{10}(E_{N^*}) \approx p_1 log_{10}(N^*) + p_2$$

$$0 \approx 1.4291(log_{10}(N^*)) - 15.6194$$

$$\frac{15.6194}{1.4291} \approx log_{10}(N^*)$$

$$10.9295 \approx log_{10}(N^*)$$

$$N^* \approx 10^{10.9295}$$

$$N^* \approx 8.5016 \times 10^{10}$$

Therefore, for a random matrix A, the size of the matrix must reach approximately $10^{10.9295}$ columns by $10^{10.9295}$ rows for the round-off error in the GE algorithm to be the same order of magnitude as the solution *x*. Since this number was approximated from random matrices, it can vary from 10^{10} to 10^{11}.

```
% Computing Assignment #2
% Author: Andy Liu
% ID: 301472847

EN1 = GERandom( 5, 500 );
EN2 = GERandom( 10, 500 );
EN3 = GERandom( 100, 500 );
EN4 = GERandom( 250, 500 );
EN5 = GERandom( 500, 500 );
EN6 = GERandom( 1000, 500 );
EN7 = GERandom( 2000, 500 );
EN8 = GERandom( 2500, 500 );
EN9 = GERandom( 3000, 500 );

EN = [EN1 EN2 EN3 EN4 EN5 EN6 EN7 EN8 EN9];
N = [5 10 100 250 500 1000 2000 2500 3000];
p = polyfit(log10(N), log10(EN), 1);
predicted_log_EN = polyval(p, log10(N));

loglog(N, EN, 'o', 'MarkerFaceColor', 'black', 'MarkerEdgeColor', 'black', 'DisplayName', 'Mean Error');
title('Mean error EN gaussian elimination');
hold on;
loglog(N, 10.^predicted_log_EN, 'r-', 'DisplayName', 'Linear Fit');
xlabel('N');
ylabel('EN');
legend('show');
```

```
function EN = GERandom( N, M )

% GERANDOM: Gaussian elimination for a random matrix.
%
%     Over a series of M trials, compute the solution to A*x = b
%     for an NxN random matrix A. Compute the mean/average error
%     over all M trial solutions and plot the results.
%
% YOU NEED MODIFY THIS CODE FOR CA3.

%N = 10;  % matrix size
%M = 100; % number of trials


delta = zeros(M,1);  % vector of errors
x = ones(N,1);       % exact solution vector

for k = 1 : M,
  A = 2*rand(N,N)-1; % random NxN matrix with entries in [-1, 1]
  b = A * x;         % compute the right-hand side vector
  y = A \ b;         % approximate solution in floating-point
  delta(k) = max(abs(x-y)); % compute max-norm error
end

% Compute average error over all trials
EN = mean(delta)
```