Kevin San Gabriel
301342241
MACM 316 D100 Benjamin Adcock
Computing Assignment #2

Case 1: Solving the linear system Ax = b with GE and QR where A is an N*N random matrix with unit normal entries and x is a vector of 1s. Number of trials chosen is 7 where each trial has a different range for N. 7 trials were chosen for computing time efficiency and the values of N were chosen to illustrate the significant increase in error and time as N grows in small and large variations.

Case 2: Solving the linear system Ax = b with GE and QR where A = B^TB and B is a random matrix with unit normal entries and x is a vector of 1s. The same number of trials and ranges of N as Case 1 were chosen to compare the difference between when A = B^TB and when A = N*N.

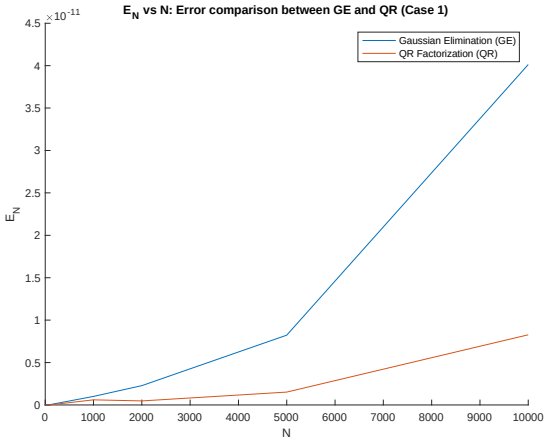

Fig 1. Robustness between QR and GE methods
in solving the linear system Ax = b in Case 1

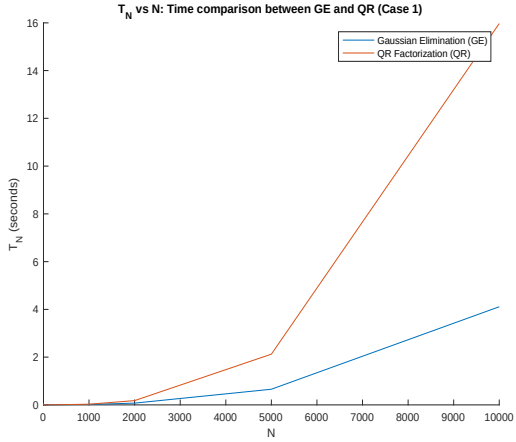

Fig 2. The difference in time it takes to solve the
linear system Ax = b between GE and QR in Case 1

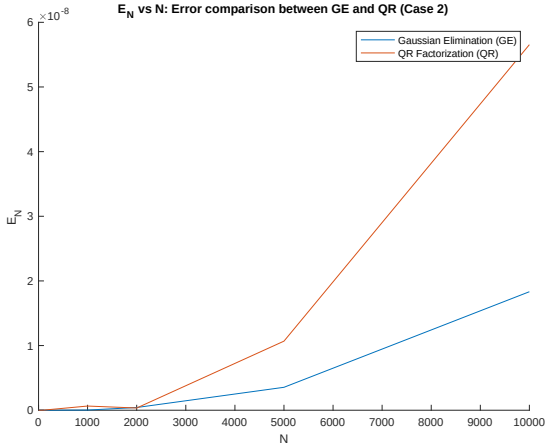

Fig. 3. Robustness between QR and GE methods
In solving the linear system Ax = b in Case 2

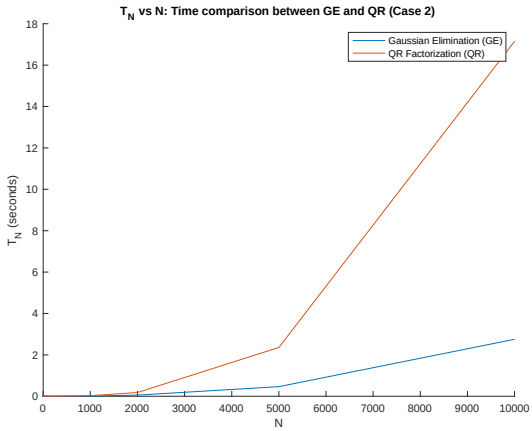

Fig. 4. Time difference in solving the linear system Ax = b
between QR and GE methods in Case 2

Notes:
-For case 1, as N grows, GE incurs error at an increasingly faster rate than QR (Fig. 1)
-For case 2, as N grows, QR incurs error at an increasingly faster rate than GE (Fig. 3)

-For case 1 & 2, as N grows, QR takes an increasingly longer time to solve Ax = b compared to GE
-At N = 5000, for cases 1 & 2, the error incurred and time taken to solve Ax = b increases significantly

Discussion: With regards to robustness, the QR method is a more stable method for solving Ax = b for reasonably sized matrices when compared to the GE method as shown in Fig. 1. The trade-off is that it is more costly in time efficiency as can be depicted in Fig. 2 and Fig. 4. QR is approximately twice as expensive in time efficiency compared to GE but this is only the case for reasonably sized matrices (N = 1 to N = 5000). Generally, GE has less approximation accuracy but is more time efficient than QR.

MATLAB Code

```
%%% Kevin San Gabriel
%%% 301342241
%%% MACM 316 D100 Benjamin Adcock
%%% Computing Assignment 1

N_vals = [1, 10, 100, 1000, 2000, 5000, 10000];

num_trials = length(N_vals);
dim = size(N_vals);


%%%
%%% Case 1
%%%

GE_error = zeros(dim);    %Array of error values for GE trials
GE_times = zeros(dim);    %Array of timing data for GE trials
QR_error = zeros(dim);    %Array of error values for QR trials
QR_times = zeros(dim);    %Array of timing data for QR trials


%Generate errors and times for GE
for i = 1:num_trials
    N_val = N_vals(i);
    A = randn(N_val);
    x = ones(N_val, 1);
    b = A*x;

    tic;
    xhat_GE = A\b;
    GE_times(i) = toc;
    GE_error(i) = max(abs(xhat_GE-x));

    tic;
    [Q,R] = qr(A);
    xhat_QR = R\(transpose(Q)*b);
    QR_times(i) = toc;
    QR_error(i) = max(abs(xhat_QR-x));
end

figure(1);
title('E_N vs N: Error comparison between GE and QR (Case 1)');
xlabel('N');
ylabel('E_N');

hold on;
plot(N_vals, GE_error);
plot(N_vals, QR_error);
hold off;
legend('Gaussian Elimination (GE)','QR Factorization (QR)');

figure(2);
```

```
title('T_N vs N: Time comparison between GE and QR (Case 1)');
xlabel('N');
ylabel('T_N (seconds)');
hold on;
plot(N_vals, GE_times);
plot(N_vals, QR_times);
hold off;
legend('Gaussian Elimination (GE)', 'QR Factorization (QR)');

%%%
%%% Case 2
%%%

%Reset our error and times storage arrays for the A = B^T*B case
GE_error = zeros(dim);
GE_times = zeros(dim);
QR_error = zeros(dim);
QR_times = zeros(dim);


%A = B^T*B case for GE
for i = 1:num_trials
    N_val = N_vals(i);
    B = randn(N_val);
    A = transpose(B)*B;
    x = ones(N_val, 1);
    b = A*x;

    tic;
    xhat_GE = A\b;
    GE_times(i) = toc;
    GE_error(i) = max(abs(xhat_GE - x));

    tic;
    [Q,R] = qr(A);
    xhat_QR = R\(transpose(Q)*b);
    QR_times(i) = toc;
    QR_error(i) = max(abs(xhat_QR - x));

end

figure(3);
title('E_N vs N: Error comparison between GE and QR (Case 2)');
xlabel('N');
ylabel('E_N');

hold on;
plot(N_vals, GE_error);
plot(N_vals, QR_error);
hold off;
legend('Gaussian Elimination (GE)','QR Factorization (QR)');

figure(4);
title('T_N vs N: Time comparison between GE and QR (Case 2)');
xlabel('N');
```

```
ylabel('T_N (seconds)');
hold on;
plot(N_vals, GE_times);
plot(N_vals, QR_times);
hold off;
legend('Gaussian Elimination (GE)', 'QR Factorization (QR)');
```