COMPUTING ASSIGNMENT 2

(a)

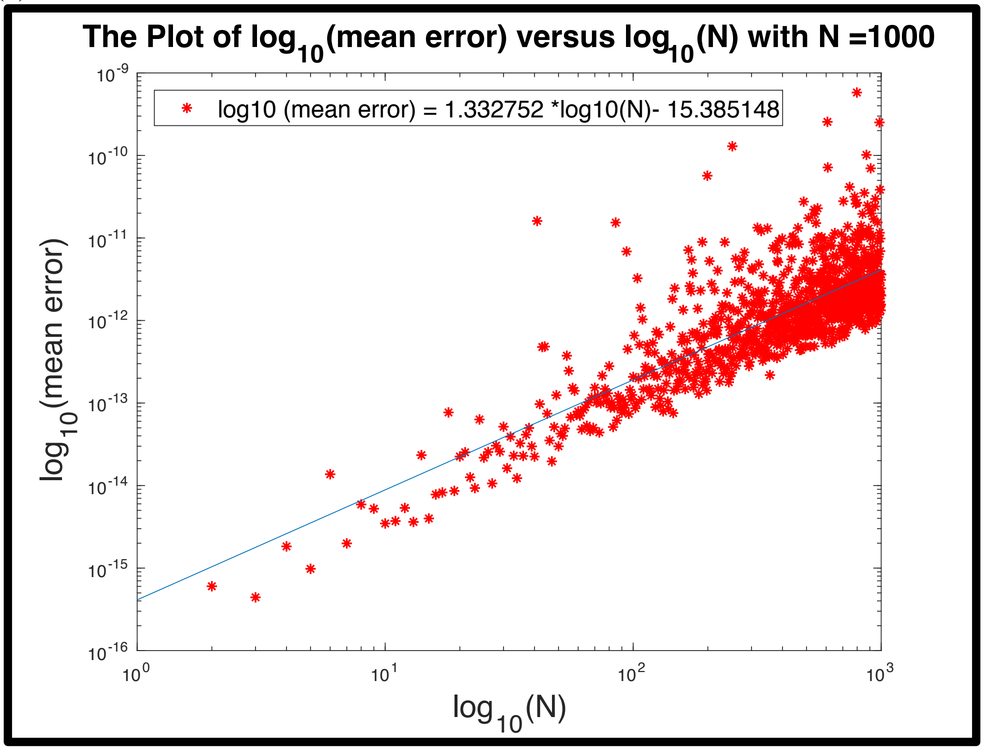

Fig.1

(b) N is the matrix size and M is the number of trials. To justify the value of N, I plotted N=1000 which is a large number to be computed by the ordinary computers. I chose N=1000 because larger matrices are more accurate but they are definitely less efficient as it takes a bit of time for them to compute. Also, if N starts from 1, we cannot get the coefficient from polyfit function as 1*1 matrix has no error (error=0). Then, log10(0) is negative infinite and there will be no linear fit to the data. To justify the value of M, I plotted the same N with different trials and calculate the estimated N*. As explained in part(c) the N* came out to be similar with different values of M which indicated that the number of trails have very little effect on N*. So, I used M=100 since it is efficient and it also does not make a big difference on using higher values of M.

(c) The aim of this assignment is to find the matrix size N=N* at which the mean error of Gaussian Elimination En ≈ 1(mean error =1). We have to extrapolate the data to find an estimated N* because it's not possible for personal or small computers to find the exact N* which requires a lot of big computations. In order to achieve this, we can plot log10 (mean error) versus log10 (N) to get the linear fit line, which is in the format:log10 (mean error) = slope*log10 (N) + intercept. The slope and intercept can be produced by polyfit function. When mean error equals to 1, log10 (N) = –intercept/slope, and N* is roughly '10^intercept/slope'.

(d) According to the numbers we got in **Fig 1**.
The average slope = 1.3327
The intercept = -15.3851
The average log10(N) = -intercept/slope = 11.5443
So, estimated N* ≈ 10^12

CODE (Fig 1)

```
clear


    N = 1000;
    M = 100;

    errs = zeros(M,1);
    n_vector = zeros(N,1);
    mean_errs = zeros(N,1);



for j=1:N
    for i=1:M
        A=spdiags(rand(j,3), -1:1, j,j); % creates a tridiagonal matrix
with random entries along each diagonal.
        x=ones(j,1);  % exact solution vector
        b = A*x;  % Compute the right-hand side vector
        z = A\b;  % Solve the linear system
        errs(i) = max(abs(z-x));  % Compute the error
    end

   n_vector(j) =j;
   mean_errs(j) = mean(errs);  %compute the mean error
end
log_nvec =log10(n_vector);
log_mean =log10(mean_errs);

loglog(n_vector, mean_errs,'r*');  %plot the scatter
hold on;
p = polyfit(log_nvec(2:end),log_mean(2:end),1); %y=ax+b, get a and b
fit =10.^(polyval(p,log10(1:N)));  %plot the linear regression
loglog(n_vector, fit)
hold off;

title(['The Plot of log_{10}(mean error) versus log_{10}(N) with N
=1000'],'fontsize',18)
xlabel('log_{10}(N)','fontsize',18);
ylabel('log_{10}(mean error)','fontsize',18);


l = legend('Location', 'northwest'); %location of the legend
l.FontSize =14;

if p(2) < 0
    operator = '-';
else
    operator = '+';
end

legend(sprintf('log10 (mean error) = %f *log10(N)%s %f', p(1),operator,
abs(p(2)))); %legend
```