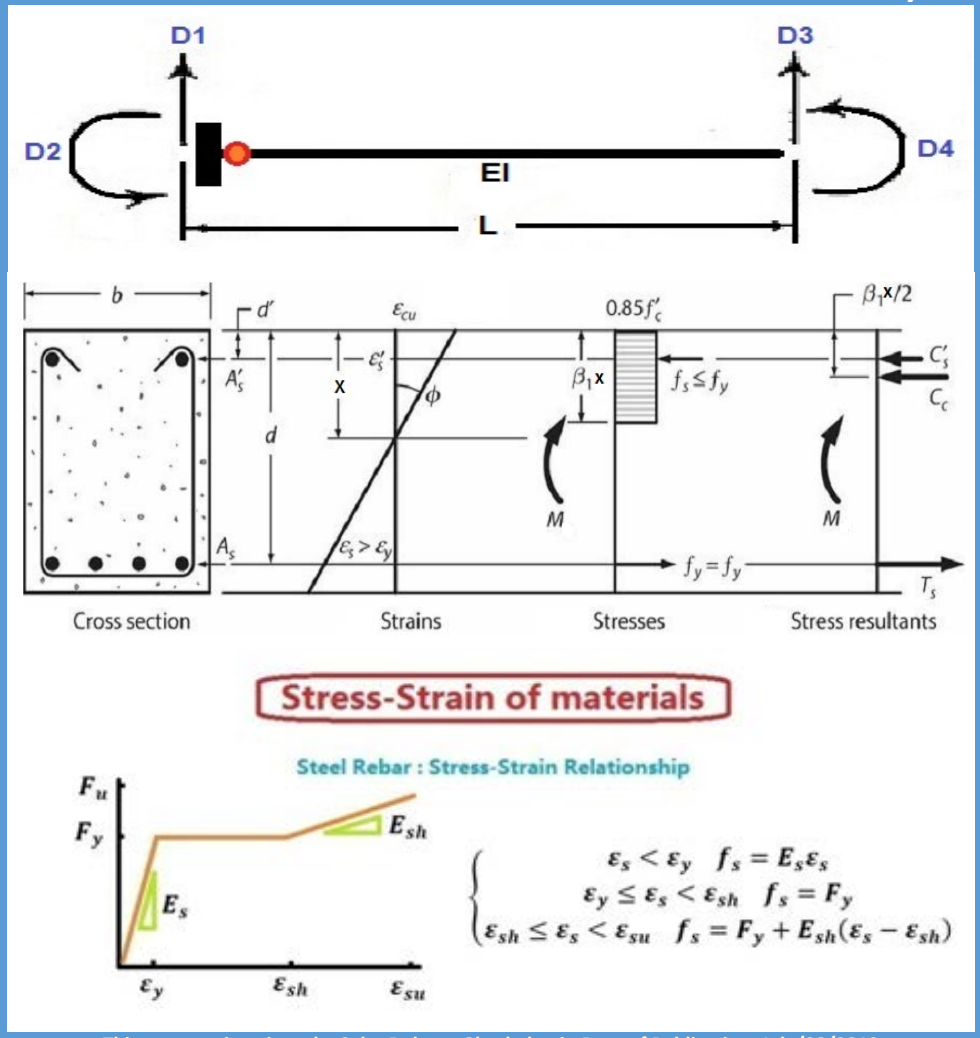
>> IN THE NAME OF GOD <<

Failure Probability Analysis Cantilever Beam with Concrete Section Based on Witney Stress Block in C Programming



This program is written by Salar Delavar Ghashghaei - Date of Publication: July/28/2019
E-mail: salar.d.ghashghaei@gmail.com

C code:

```
#include <stdio.h>
#include <windows.h> // text color
#include <math.h>
#define N 10
#define Pi 3.14159265358979238462643
#define FileName1 "FailureProbabilityAnaCantileverBeam-inputDATA01.csv"
#define FileName2 "FailureProbabilityAnaCantileverBeam-inputDATA02.csv"
#define FileName3 "Graph-outputHTML.html"
#define FileName4 "outputEXCEL.csv"
double ABS(double B);
double MAX_ABS(double A[],int n);
double SQRT2(double D,double tolerance,int itermax);
double cnd_manual(double x);
void IMPORT_DATA01(double data[],int &n);
void IMPORT_DATA02(double b[],double h[],double As1[],double As2[],double d1[],double d2[],double ecu[],double fc[],double esh[],double esu[],double Es[],double fy[],double fu[],int &m);
void MessageCheck_IMPORT_DATA02(double b[],double h[],double As1[],double As2[],double d1[],double d2[],double ecu[],double fc[],double esh[],double esu[],double Es[],double fy[],double fu[],int m);
void MessageCheck_IMPORT_DATA01(double data[],int n);
void MessageInitialData(double data[],int n,double b[],double h[],double As1[],double As2[],double d1[],double d2[],double ecu[],double fc[],double esh[],double esu[],double Es[],double fy[],double fu[],int m);
void MessageInputDataTEXT();
void MessageErrorReportTEXT();
void MessageAnalysisReport();
void textcolor(int ForgC);
void ANALYSIS_Probability(double data[],double MomUU[],int m);
void ANALYSIS_Moment(double data[],double b[],double h[],double As1[],double As2[],double d1[],double d2[],double ecu[],double fc[],double esh[],double esu[],double Es[],double fy[],double fu[],int m,double MomUU[]);
double AVERAGE(double DATA[],int n);
double VARIANCE(double DATA[],double average,int n);
double STANDARD_DEVIATION(double variance);
void OUTPUT_HTML_GRAPH(double X[],double Y[],int n,const char text1[],const char text2[],const char text3[]);
void OUTPUT_EXCEL(double A[],double B[],int n);
int main(){
//Normal Distribution = ND
//Standard_Deviation =SD
//RSD = (Ratio Standard Deviation) =RSD = SD/ND
//ND = Load Normal Distribution
//ND = Resistance Normal Distribution
//CSF = Central safety factor
//COV = Coefficient of variation
double b[N],h[N],As1[N],As2[N],d1[N],d2[N],ecu[N],fc[N],esh[N],esu[N],Es[N],fy[N],fu[N],MomUU[N];
double result,data[4];
int n,m,i;
IMPORT_DATA01(data,n);
MessageCheck_IMPORT_DATA01(data,n);
IMPORT_DATA02(b,h,As1,As2,d1,d2,ecu,fc,esh,esu,Es,fy,fu,m);
MessageCheck_IMPORT_DATA02(b,h,As1,As2,d1,d2,ecu,fc,esh,esu,Es,fy,fu,m);
textcolor(14);
MessageInitialData(data,n,b,h,As1,As2,d1,d2,ecu,fc,esh,esu,Es,fy,fu,m);
textcolor(11);
ANALYSIS_Moment(data,b,h,As1,As2,d1,d2,ecu,fc,esh,esu,Es,fy,fu,m,MomUU);
ANALYSIS_Probability(data,MomUU,m);
system("pause");
return 0;
}
void IMPORT_DATA01(double data[],int &n){
int i = 0;
FILE *InputFile;
InputFile = fopen(FileName1, "r");
if (!InputFile){
MessageErrorReportTEXT();
printf("        File is not available! -> [%s] \n",FileName1);
Sleep(6000);
exit(1);
```

```
}
char line[1000];
do{
fscanf(InputFile,"%lf",&data[i]);
//printf("%e,%e\n",ND[i],RSD[i]);
i++;
}
while(i < 4 && fgets(line,sizeof(line),InputFile) != NULL);
n = i;
//printf("%d\n",n);
}
void IMPORT_DATA02(double b[],double h[],double As1[],double As2[],double d1[],double d2[],double ecu[],double fc[],double esh[],double esu[],double Es[],double fy[],double fu[],int &m){
int i = 0;
FILE *InputFile;
InputFile = fopen(FileName2, "r");
if (!InputFile){
MessageErrorReportTEXT();
printf("         File is not available! -> [%s] \n",FileName2);
Sleep(6000);
exit(1);
}
char line[1000];
do{
fscanf(InputFile,"%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf",&b[i],&h[i],&As1[i],&As2[i],&d1[i],&d2[i],&ecu[i],&fc[i],&esh[i],&esu[i],&Es[i],&fy[i],&fu[i]);
//printf("%.3e %.3e %.3e %.3e %.3e %.3e %.3e %.3e %.3e %.3e %.3e %.3e %.3e %.3e %.3e\n",b[i],h[i],As1[i],As2[i],d1[i],d2[i],ecu[i],fc[i],esh[i],esu[i],Es[i],fy[i],fu[i]);
i++;
}
while(i < 100 && fgets(line,sizeof(line),InputFile) != NULL);
m = i-1;
//printf("%d\n",m);
}
double ABS(double B){
if (B < 0)
B = -B;//Absolute number
else
B = B;
return B;
}
double SQRT2(double D,double tolerance,int itermax){
int it;
double residual,x,dx,dx_ABS,f,df;
it = 0; // initialize iteration count
residual = 100; // initialize residual
x = 1;// initialize answer
while (residual > tolerance){
f = x*x - D;
df = 2 * x;
dx = f/df;
x -= dx;
residual = ABS(dx); // abs residual
it += 1; // increment iteration count
//printf("f: %f -\tdx: %f -\tresidual: %f\n",f,dx,residual);
if (it == itermax){
//printf("\tSQRT2(number,power) : SQRT2(%f) - iteration: %d ->   ## The solution is not converged ##\n",D,it);
break;
}
}
if (it < itermax){
//printf("\tSQRT(number,power) - SQRT(%f,%f) : %f \n",D,n, x);
return x;
}
}
double cnd_manual(double x){
```

```
double L, K, w ;
/* constants */
double const a1 = 0.31938153, a2 = -0.356563782, a3 = 1.781477937;
double const a4 = -1.821255978, a5 = 1.330274429;

L = fabs(x);
K = 1.0 / (1.0 + 0.2316419 * L);
w = 1.0 - 1.0 / sqrt(2 * Pi) * exp(-L *L / 2) * (a1 * K + a2 * K *K + a3 * pow(K,3) + a4 * pow(K,4) + a5 * pow(K,5));

if (x < 0 ){
w= 1.0 - w;
}
return w;
}
void MessageInputDataTEXT(){
int i;
char Ql=176;
printf("\n     ");
for (i=1;i<50;i++)
printf("%c",Ql);
printf(" Input Data ");
for (i=1;i<50;i++)
printf("%c",Ql);
printf("\n");
}
void MessageAnalysisReport(){
int i;
char Ql=176;
printf("\n     ");
for (i=1;i<50;i++)
printf("%c",Ql);
printf(" Analysis Report ");
for (i=1;i<50;i++)
printf("%c",Ql);
printf("\n");
}
void MessageErrorReportTEXT(){
int i;
char Ql;
Ql=176;
textcolor(12);
printf("\a\n     ");
for (i=1;i<50;i++)
printf("%c",Ql);
printf(" Error Report ");
for (i=1;i<50;i++)
printf("%c",Ql);
printf("\n");
}
void textcolor(int ForgC){
WORD wColor;
//This handle is needed to get the current background attribute

HANDLE hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
CONSOLE_SCREEN_BUFFER_INFO csbi;
//csbi is used for wAttributes word

if(GetConsoleScreenBufferInfo(hStdOut, &csbi)){
//To mask out all but the background attribute, and to add the color
wColor = (csbi.wAttributes & 0xF0) + (ForgC & 0x0F);
SetConsoleTextAttribute(hStdOut, wColor);
}
return;
```

```
}
void MessageCheck_IMPORT_DATA01(double data[],int n){
for (int i=0;i<n;i++){
if (data[i] <= 0){
MessageErrorReportTEXT();
printf("            Please check this file! -> [ FailureProbabilityAnalSMomentMULTI-inputDATA.csv ]\n");
printf("                        *** Negative or Zero data input value is not acceptable ***\n");
printf("\t  Maximum number of iterations:                  %.3e\n",data[0]);
printf("\t  Tolerance for convergence:                     %.3e\n",data[1]);
printf("\t  Length of Beam:                              %.3e\n",data[2]);
printf("\t  Coefficient of variation:                      %.3e\n",data[3]);
Sleep(40000);
exit(1);
}
}
}
void MessageCheck_IMPORT_DATA02(double b[],double h[],double As1[],double As2[],double d1[],double d2[],double ecu[],double fc[],double esh[],double esu[],double Es[],double fy[],double fu[],int m){
for (int i=0;i<m;i++){
if (b[i] <= 0 || h[i] <= 0 || As1[i] <= 0 || As2[i] <= 0 || d1[i] <= 0 || d2[i] <= 0 || ecu[i] <= 0 || fc[i] <= 0 || esh[i] <= 0 || esu[i] <= 0 || Es[i] <= 0 || fy[i] <= 0 || fu[i] <= 0){
MessageErrorReportTEXT();
printf("            Please check this file! -> [ %s ]\n",FileName2);
printf("                        *** Negative or Zero data input value is not acceptable ***\n");
printf("\t  Width of Section:                %.3e\n",b[i]);
printf("\t  Height of Section:               %.3e\n",h[i]);
printf("\t  Rebar Area 1:                   %.3e\n",As1[i]);
printf("\t  Rebar Area 2:                   %.3e\n",As2[i]);
printf("\t  Rebar Distance 1:                %.3e\n",d1[i]);
printf("\t  Rebar Distance 2:                %.3e\n",d2[i]);
printf("\t  Concrete Ultimate Strain:         %.3e\n",ecu[i]);
printf("\t  Concrete strength:               %.3e\n",ecu[i]);
printf("\t  Rebar Strain at strain-hardening:   %.3e\n",esh[i]);
printf("\t  Rebar Ultimate steel strain:       %.3e\n",esu[i]);
Sleep(40000);
exit(1);
}
}
}
void MessageInitialData(double data[],int n,double b[],double h[],double As1[],double As2[],double d1[],double d2[],double ecu[],double fc[],double esh[],double esu[],double Es[],double fy[],double fu[],int m){
char Qa,Qb,Qc,Qd,Qe,Qf,Qg,Qk;
int  i;
Qa=201;Qb=205;Qc=187;Qd=200;Qe=188,Qf=186,Qg=204,Qk=185;
printf("\t\t\t\t%c",Qa);
for (i=1;i<71;i++)
printf("%c",Qb);
printf("%c\n",Qc);
printf("\t\t\t\t%c                    >> IN THE NAME OF GOD <<                %c\n",Qf,Qf);
printf("\t\t\t\t%c Failure Probability Analysis Cantilever Beam with Concrete Section   %c\n",Qf,Qf);
printf("\t\t\t\t%c                  Based on Witney Stress Block              %c\n",Qf,Qf);
printf("\t\t\t\t%c                    UNIT: [Newton-Millimeter]               %c\n",Qf,Qf);
printf("\t\t\t\t%c",Qg);
for (i=1;i<71;i++)
printf("%c",Qb);
printf("%c\n",Qk);
printf("\t\t\t\t%c         This program is written by Salar Delavar Ghashghaei       %c\n",Qf,Qf);
printf("\t\t\t\t%c              E-mail: salar.d.ghashghaei@gmail.com             %c\n",Qf,Qf);
printf("\t\t\t\t%c",Qg);
for (i=1;i<71;i++)
printf("%c",Qb);
printf("%c\n",Qk);
printf("\t\t\t\t%c          Notice: All input values must be positive              %c\n",Qf,Qf);
printf("\t\t\t\t%c      _   _____________________________________________              %c\n",Qf,Qf);
printf("\t\t\t\t%c      |   |                                 |            %c\n",Qf,Qf);
printf("\t\t\t\t%c          |     #                     #      |            %c\n",Qf,Qf);
```

```
printf("\t\t\t\t%c          |     #                    #      |             %c\n",Qf,Qf);
printf("\t\t\t\t%c    Width |     # As1               As2 #      |              %c\n",Qf,Qf);
printf("\t\t\t\t%c          |     #                    #      |             %c\n",Qf,Qf);
printf("\t\t\t\t%c      |   |     #                    #      |              %c\n",Qf,Qf);
printf("\t\t\t\t%c      _   |_____________________________________________|             %c\n",Qf,Qf);
printf("\t\t\t\t%c          |<-d1->|                                        %c\n",Qf,Qf);
printf("\t\t\t\t%c          |<-            d2          ->|                  %c\n",Qf,Qf);
printf("\t\t\t\t%c          |<-              Height            ->|            %c\n",Qf,Qf);
printf("\t\t\t\t%c",Qd);
for (i=1;i<71;i++)
printf("%c",Qb);
printf("%c\n",Qe);
MessageInputDataTEXT();
printf("\t  Maximum number of iterations:                  %.3e\n",data[0]);
printf("\t  Tolerance for convergence:                     %.3e\n",data[1]);
printf("\t  Length of Beam:                               %.3e\n",data[2]);
printf("\t  Coefficient of variation:                       %.3e\n",data[3]);
}
void ANALYSIS_Probability(double data[],double MomUU[],int m){
int it,itermax;
double L,average,variance,standard_deviation,FP[100],Cr_load[100];
double x=100,dx,f,df,residual,tolerance,esp,fp=.01;
average = AVERAGE(MomUU,m);
variance = VARIANCE(MomUU,average,m);
standard_deviation = STANDARD_DEVIATION(variance);
printf("\t  Moment Capacity Average:                       %.3e\n",average);
printf("\t  Moment Capacity Sample Standard Deviation:        %.3e\n",standard_deviation);
double reliability_index,COV;
itermax = data[0];
tolerance = data[1];
L = data[2];
COV = data[3];
for(int i=0;i<100;i++){
FP[i] = fp*(i+1);
reliability_index = 1/(cnd_manual(1-FP[i]));
residual = 100; // initialize residual
it = 0;
while (residual > tolerance){// Newton Method
f = (average-x)*(average-x)-((standard_deviation*standard_deviation)+(COV*x*COV*x))*reliability_index*reliability_index;
df = - 2*average + 2*x - 2*COV*COV*x*reliability_index*reliability_index;
dx = f/df;
x -= dx;// update x
residual = ABS(dx); // evaluate residual
it += 1; // increment iteration count
if (it == itermax){ // stop the the analysis of this step please of Convergence
//x = 100000;
MessageErrorReportTEXT();
printf("\t\t\t\t\t\t\t\t Solution Not Found!\n");
textcolor(11);
//break;
}
}//while
if (it < itermax)// iteration control
Cr_load[i] = x/L;
printf("\t  iteration: %d \n",i);
printf("\t  Failure Probability [%d]:                    %.3e  percent\n",i+1,FP[i]*100);
printf("\t  Safety Probability [%d]:                     %.3e  percent\n",i+1,100-FP[i]*100);
printf("\t  Reliability Index [%d]:                     %.3e \n",i+1,reliability_index);
printf("\t  Critical Load [%d]:                         %.3e \n\n",i+1,Cr_load[i]);
}
OUTPUT_HTML_GRAPH(FP,Cr_load,m-1,"Failure Probability Analysis Cantilever Beam","Failure Probability","Critical Load");
OUTPUT_EXCEL(FP,Cr_load,m);
system("start /w Graph-outputHTML.html");
```

```
}
void ANALYSIS_Moment(double data[],double b[],double h[],double As1[],double As2[],double d1[],double d2[],double ecu[],double fc[],double esh[],double esu[],double Es[],double fy[],double fu[],int m,double MomUU[]){
double ey,Esh,x,beta1,tolerance,a,ec,Fsum;
double residual,A,A_tan,Cc,Cctan,FsSUM,FstanSUM,SUM,dx;
double MomU;
int z,it,itermax;
double fs,fstan,Fs1,Fstan1,Fs2,Fstan2,es;
itermax = data[0];
tolerance = data[1];
MessageAnalysisReport();
printf("   --------------------------------\n");
printf("    Section   Ulitmate-Moment   \n");
printf("   --------------------------------\n");

for (z=0; z<m; z++){
ey = fy[z]/Es[z];
Esh = (fu[z]-fy[z])/(esu[z]-esh[z]);
// Ultimate capacity
it = 0; // initialize iteration count
residual = 100; // initialize residual
x = .25*h[z];// initialize nuteral axis
if (fc[z]<= 30)
{beta1=0.85;}
else if (fc[z]> 30 && fc[z]< 55)
{beta1=0.85-.008*(fc[z]-30);}
else  if (fc[z]>= 55)
{beta1=0.65;}
while (residual > tolerance){
a=beta1*x;Cc=0.85*fc[z]*a*b[z];Cctan=0;ec=0.5*(h[z]-a);
SUM = 0;
FsSUM = 0;
es = (ecu[z]*(d1[z]-x))/x;
if  (es>=-ey && es<=ey)
{fs=Es[z]*es;fstan=(Es[z]*ecu[z]*d1[z]/(x*x));}
else if (es>ey && es<=esh[z])
{fs=fy[z];fstan=0;}
else if (es<-ey && es>=-esh[z])
{fs=-fy[z];fstan=0;}
else if (es>esh[z] && es<=esu[z])
{fs=fy[z]+Esh*(abs(es)-esh[z]);fstan=(Esh*ecu[z]*d1[z]/(x*x));}
else if (es<-esh[z] && es>=-esu[z])
{fs=-fy[z]-Esh*(abs(es)-esh[z]);fstan=(Esh*ecu[z]*d1[z]/(x*x));}
else if (es<-esu[z] || es>esu[z])
{fs=0;fstan=0;}

if (d1[z]>a)
{Fs1 = As1[z]*fs;Fstan1 = As1[z]*fstan;}
else if (d1[z]<=a)
{Fs1 = As1[z]*(fs-0.85*fc[z]);Fstan1 = As1[z]*(fstan-0.85*fc[z]);}

FsSUM += Fs1;
FstanSUM += Fstan1;
SUM += Fs1*d1[z]-0.5*h[z];

es = (ecu[z]*(d2[z]-x))/x;
if  (es>=-ey && es<=ey)
{fs=Es[z]*es;fstan=(Es[z]*ecu[z]*d2[z]/(x*x));}
else if (es>ey && es<=esh[z])
{fs=fy[z];fstan=0;}
else if (es<-ey && es>=-esh[z])
{fs=-fy[z];fstan=0;}
else if (es>esh[z] && es<=esu[z])
{fs=fy[z]+Esh*(abs(es)-esh[z]);fstan=(Esh*ecu[z]*d2[z]/(x*x));}
```

```
else if (es<-esh[z] && es>=-esu[z])
{fs=-fy[z]-Esh*(abs(es)-esh[z]);fstan=(Esh*ecu[z]*d2[z]/(x*x));}
else if (es<-esu[z] || es>esu[z])
{fs=0;fstan=0;}

if (d2[z]>a)
{Fs2=As2[z]*fs;Fstan2=As2[z]*fstan;}
else if (d2[z]<=a)
{Fs2=As2[z]*(fs-0.85*fc[z]);Fstan2=As2[z]*(fstan-0.85*fc[z]);}

FsSUM += Fs2;
FstanSUM += Fstan2;
SUM += Fs2*d2[z]-0.5*h[z];
A = Cc-FsSUM;
A_tan = Cctan-FstanSUM;
dx = A/A_tan;
residual = ABS(dx); // evaluate residual
//    printf("\t\t x: %f \n",x);
//    printf("\t\t : %f \n",fs);
//    printf("\t\t A: %f \n",A);
//    printf("\t\t A_tan: %f \n",A_tan);
//    printf("\t\t dx: %f \n",dx);
//    printf("\t\t residual: %.3e \n\n",residual);
it += 1; // increment iteration count
x -= dx; // update x
if (it == itermax){ // stop the the analysis of this step please of Convergence
printf("\n      Iteration reached to Ultimate %d  - strain: %f  - residual: %f",it,ecu[z],residual);
printf("       ## The solution for this step is not converged. Please check your model ##");
exit(2);
//getch();
}
}// while
MomU = SUM + Cc*ec;
if (it < itermax){// iteration control
MomUU[z] = MomU;
printf("       %d         %.5e        \n",z+1,MomUU[z]);
}
}//for
}
double AVERAGE(double DATA[],int n){
double sum;
sum=0;
for (int i=0;i<n;i++)
sum += DATA[i];
return sum /n;
}
double VARIANCE(double DATA[],double average,int n){
double sum;
sum=0;
for (int i=0;i<n;i++)
sum += (DATA[i]-average)*(DATA[i]-average);
return sum /n;
}
double STANDARD_DEVIATION(double variance){
return SQRT2(variance,.000000001,10000000);
}
void OUTPUT_HTML_GRAPH(double X[],double Y[],int n,const char text1[],const char text2[],const char text3[]){
// HTML GRAPH OUTPUT
int i;
double x,y,Xmax,Ymax;
double *Xnew = new double [N];
double *Ynew = new double [N];
double *NorX = new double [N];
```

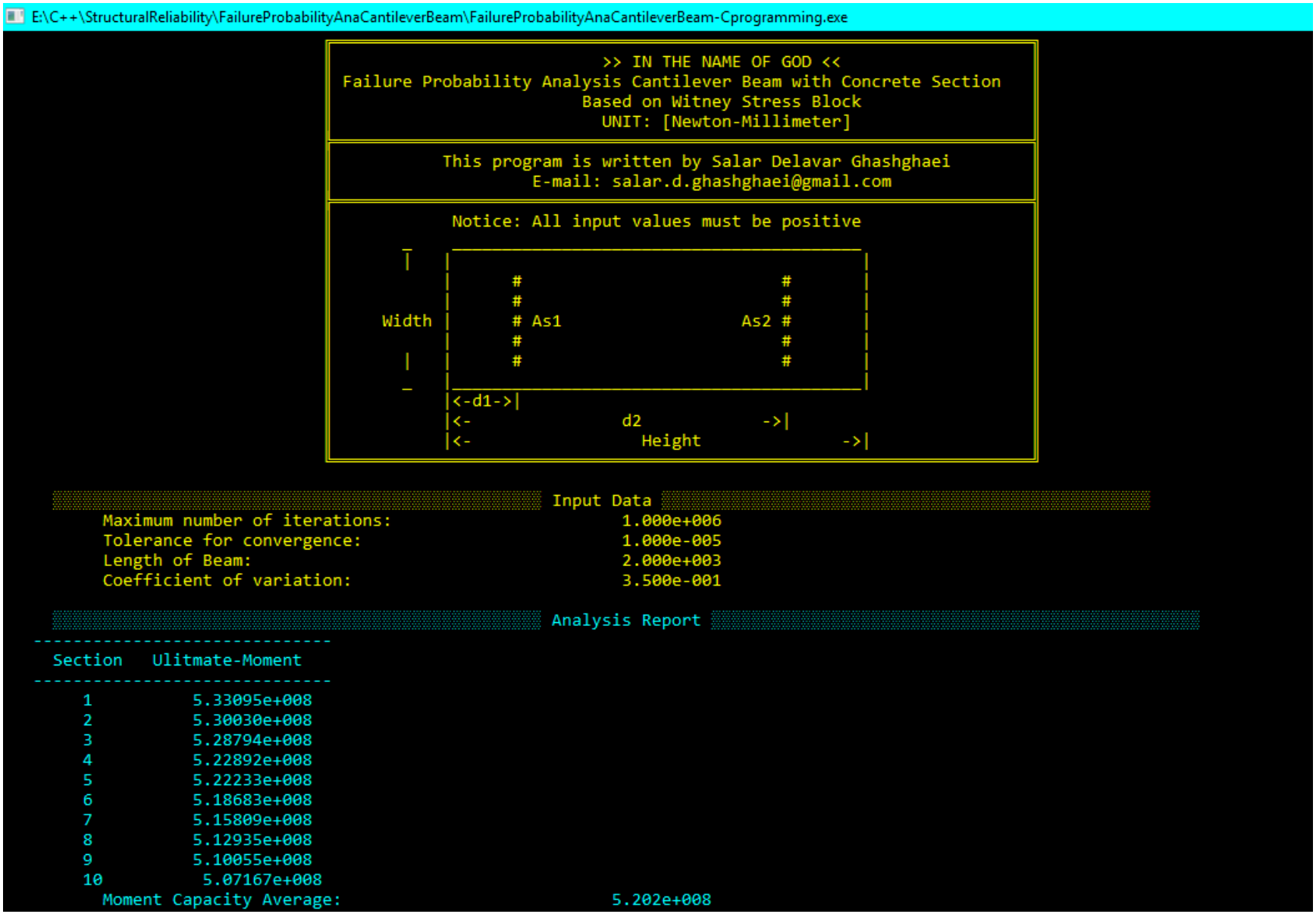
```
double *NorY = new double [N];
Xmax=MAX_ABS(X,n);
Ymax=MAX_ABS(Y,n);
Xnew[0]=0;Ynew[0]=0;
for (i=0;i<n;i++){
Xnew[i+1] = ABS(X[i]);
Ynew[i+1] = ABS(Y[i]);
}
for (i=0;i<n+1;i++){
NorX[i] = Xnew[i]/Xmax;
NorY[i] = Ynew[i]/Ymax;
//printf("\t %f   %f    \n",NorX[i],NorY[i]);
}
FILE *OutputFile;
OutputFile = fopen(FileName3, "w");
fprintf(OutputFile,"<!DOCTYPE HTML><html><body style=\"background-color:black;\"><font color=\"white\"><head><script> \n");
fprintf(OutputFile,"window.onload = function(){ \n");
fprintf(OutputFile,"var canvas = document.getElementById(\"myCanvas\");var s1 = canvas.getContext(\"2d\");var s2 = canvas.getContext('2d'); \n");
fprintf(OutputFile,"var s3 = canvas.getContext(\"2d\");var s4 = canvas.getContext(\"2d\");var s5 = canvas.getContext(\"2d\"); \n");
fprintf(OutputFile,"var x=120,y=80,X,Y,Lx=1100,Ly=500,i; \n");
fprintf(OutputFile,"s3.beginPath();s3.lineWidth = 3;s3.strokeStyle = \"cyan\";s3.rect(x,y,Lx,Ly); \n");
fprintf(OutputFile,"for(i=0;i<9;i++){s3.moveTo(x+Lx*(i+1)*.1,y+Ly);s3.lineTo(x+Lx*(i+1)*.1,y+Ly-10);}; \n");
fprintf(OutputFile,"for(i=0;i<9;i++){s3.moveTo(x,y+Ly*(i+1)*.1);s3.lineTo(x+10,y+Ly*(i+1)*.1);};s3.stroke();\n");
fprintf(OutputFile,"s1.beginPath();s1.lineWidth = 3;s1.strokeStyle = \"yellow\"; \n");
for (i=0;i<n;i++){
fprintf(OutputFile,"s1.moveTo(%f,%f);",120+NorX[i]*1100,80+500-NorY[i]*500);
fprintf(OutputFile,"s1.lineTo(%f,%f); \n",120+NorX[i+1]*1100,80+500-NorY[i+1]*500);
}
fprintf(OutputFile,"s1.stroke(); \n");
fprintf(OutputFile,"s2.beginPath();s2.lineWidth = 1;s2.strokeStyle = \"cyan\";s2.setLineDash([5, 5]); \n");
fprintf(OutputFile,"for(i=0;i<19;i++){s2.moveTo(x+Lx*(i+1)*.05,y);s2.lineTo(x+Lx*(i+1)*.05,y+Ly);} \n");
fprintf(OutputFile,"s2.lineWidth = 1;s2.strokeStyle = \"cyan\";for(i=0;i<19;i++){s2.moveTo(x,y+Ly*(i+1)*.05);s2.lineTo(x+Lx,y+Ly*(i+1)*.05);} s2.stroke();\n");
fprintf(OutputFile,"X=x+.25*Lx;Y=.7*y;s4.translate(X,Y);s4.font=\"45px serif\";s4.fillStyle = \"#7fff00\";s4.fillText(\"%s\",0,0); \n",text1);
fprintf(OutputFile,"s4.save();X=-X+.2*x;Y=-Y+y+.6*Ly;s4.translate(X,Y);s4.rotate(3*Math.PI/2);s4.font=\"15px serif\"; \n");
fprintf(OutputFile,"s4.fillStyle = \"#7fff00\";s4.textAlign = \"left\";s4.fillText(\"%s\",0,0);s4.restore(); \n",text3);
fprintf(OutputFile,"s4.save();X=.2*Lx;Y=y+Ly-20;s4.translate(X,Y);s4.rotate(2*Math.PI);s4.font=\"15px serif\";s4.fillStyle = \"#7fff00\"; \n");
fprintf(OutputFile,"s4.textAlign = \"left\";s4.fillText(\"%s\",0,0);s4.restore(); \n",text2);
for(i=0;i<10;i++){
x=.1*(i+1)*Xmax;
fprintf(OutputFile,"s5.save();X=-.29*Lx+Lx*(%d+1)*.1;Y=.3*y+Ly+20;s5.rotate(2*Math.PI);s5.font=\"16px serif\"; \n",i);
fprintf(OutputFile,"s5.fillStyle = \"#7fff00\";s5.textAlign = \"left\";s5.fillText(\"%.3e\",X,Y);s5.restore(); \n",x);
}
for(i=0;i<10;i++){
y=.1*(i+1)*Ymax;
fprintf(OutputFile,"s5.save();X=-.28*Lx-50;Y=Ly+.3*y-Ly*(%d+1)*.1;s5.rotate(2*Math.PI);s5.font=\"16px serif\"; \n",i);
fprintf(OutputFile,"s5.fillStyle = \"#7fff00\";s5.textAlign = \"left\";s5.fillText(\"%.3e\",X,Y);s5.restore(); \n",y);
}
fprintf(OutputFile,"s5.save();X=-.25*Lx;Y=.3*y+Ly+20;s5.rotate(2*Math.PI);s5.font=\"16px serif\";s5.fillStyle = \"#7fff00\";s5.fillText(0,X,Y);s5.restore(); \n");
fprintf(OutputFile,"s5.save();X=-.25*Lx-50;Y=Ly+.3*y;s5.rotate(2*Math.PI);s5.font=\"16px serif\";s5.fillStyle = \"#7fff00\";s5.textAlign = \"left\";s5.fillText(0,X,Y);s5.restore();}; \n");
fprintf(OutputFile,"</script></head><body><canvas id=\"myCanvas\" width=\"1300\" height=\"1300\" style=\"border:1px solid black;\"></canvas></body></html> \n");
fclose(OutputFile);
}
double MAX_ABS(double A[],int n){
int i;
double B[N];
double Amax;
// abs value
for (i=0;i<n;i++){
B[i] = A[i];
if(B[i] < 0)
B[i] = -B[i];
}
// Max of abs
```

```
Amax = B[0];
for (i=1;i<n;i++){
if(Amax < B[i])
Amax = B[i];
}
return Amax;
}
void OUTPUT_EXCEL(double A[],double B[],int n){
// EXCEL OUTPUT
int i;
double Bb[N];
FILE *OutputFile;
OutputFile = fopen(FileName4, "w");
fprintf(OutputFile,"          ### Output Failure Probability Analysis Cantilever Beam ###\n");
fprintf(OutputFile,"Number,Failure Probability,Critical Load\n");
for(i=0;i<n;i++){
fprintf(OutputFile,"%d,%f,%f\n",i+1,A[i],B[i]);
}
fclose(OutputFile);
}
```

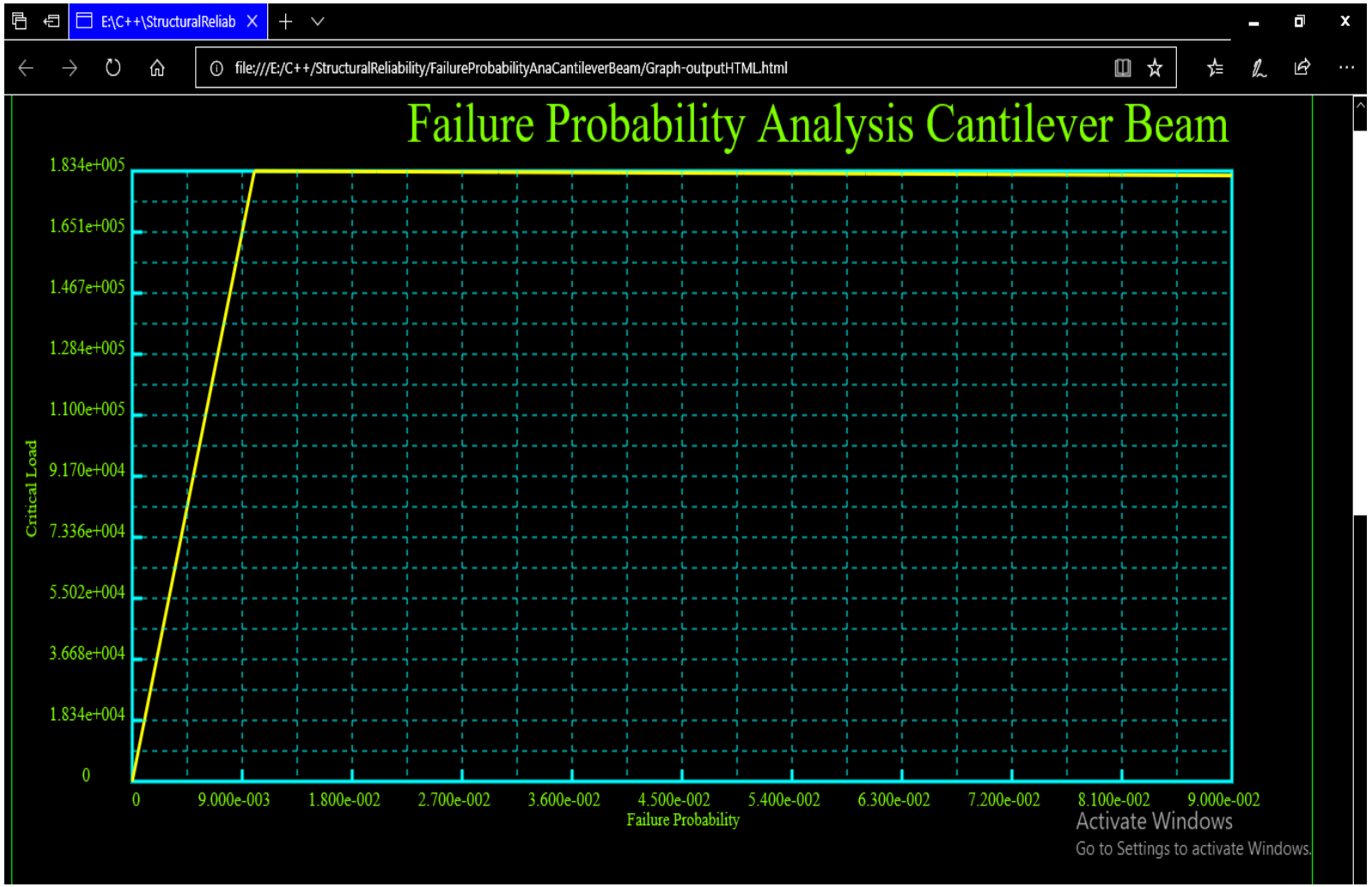
Figure(1) Input csv file

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	500	500	2500	2500	50	450	0.003	25.3	0.01	0.09	200000	400.5	598.5
2	500	500	2500	2500	50	450	0.0031	24.8	0.0098	0.085	200152	400.1	587.6
3	500	500	2500	2500	50	450	0.00298	24.95	0.0099	0.087	190123	398.25	601.1
4	500	500	2500	2500	50	450	0.0035	25.1	0.0098	0.084333	200114	397.3667	598.3333
5	500	500	2500	2500	50	450	0.00325	24.925	0.00975	0.082833	195075.5	396.2417	599.6333
6	500	500	2500	2500	50	450	0.003436	24.88	0.0097	0.081333	194046.8	395.1167	600.9333
7	500	500	2500	2500	50	450	0.003526	24.835	0.00965	0.079833	193018.1	393.9917	602.2333
8	500	500	2500	2500	50	450	0.003616	24.79	0.0096	0.078333	191989.4	392.8667	603.5333
9	500	500	2500	2500	50	450	0.003706	24.745	0.00955	0.076833	190960.7	391.7417	604.8333
10	500	500	2500	2500	50	450	0.003796	24.7	0.0095	0.075333	189932	390.6167	606.1333

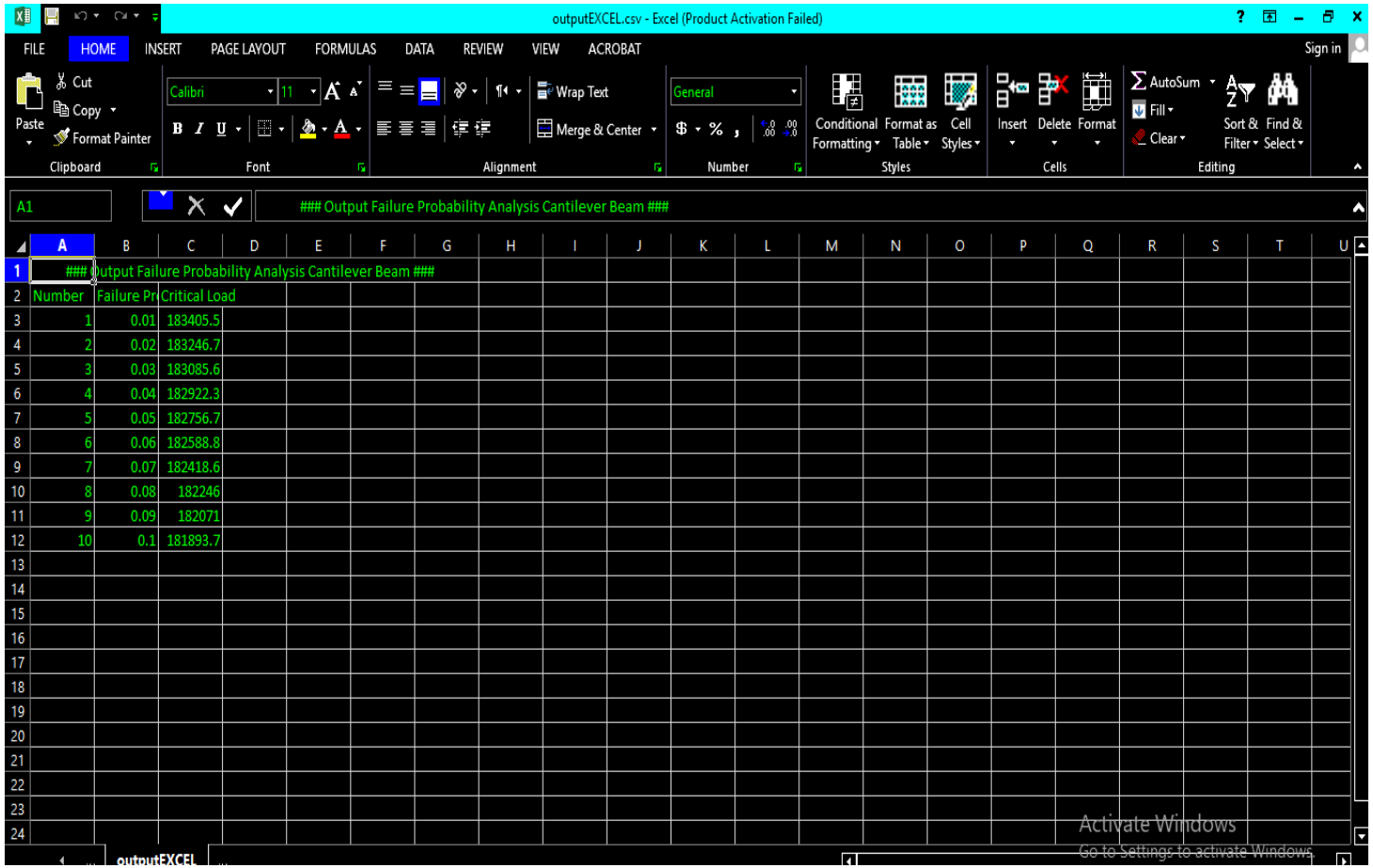
Figure(2) Input csv file



Figure(3) Analysis file



Figure(4) Analysis Output Graph



Figure(5) Analysis Output Excel