1 Introdução

Neste trabalho serão apresentadas funções programadas em Matlab para o cálculo da confiabilidade de funções de variáveis aleatórias pelos métodos First Order Reliability Method (FORM) e Monte Carlo (MC).

Os exemplos 6.7 a 6.10 do livro Probability Concepts in Engineering Planning and Design (Ang & Tang, 1996) serão aplicados para demostrar a funcionalidade das funções desenvolvidas.

2 Método FORM

O método FORM foi implementado em uma função que recebe como argumentos a função de estado limite, as médias, os desvios padrão e as distribuições das variáveis aleatórias. O gradiente da função de estado limite é calculado por diferenças finitas com passo de 10^{-3} no espaço reduzido.

form.m

```
% form(g,m,s,R,dists)
%
% Entradas:
%   g: handle para a função de estado limite
%   m: vetor de médias
%   s: vetor de desvios padrão
%   R: matriz de correlação
%   dists: nome das distribuições
% Saídas:
%   mpfp: estrutura contendo o numero de iterações (k), o ponto de falha mais
%         provável (x), o beta (beta) e a probabilidade de falha (Pf).
function mpfp = form(g,m,s,R,dists)
S = diag(s); % matriz S
meq = zeros(length(m),1);
Seq = zeros(size(S));
% para cada variavel calcular media e desvio padrão normal equivalente
for i=1:length(m)
    if strcmp(dists{i},'normal')
        Seq(i,i)=S(i,i);
        meq(i)=m(i);
    elseif strcmp(dists{i},'lognormal')
        [meq(i) Seq(i,i)] = eqLN(m(i),m(i),S(i,i));
    elseif strcmp(dists{i},'gumbel')
        [meq(i) Seq(i,i)] = eqT1(m(i),m(i),S(i,i));
    end
end
C = Seq*R*Seq;        % Matriz de covariancia
L = chol(C,'lower'); % Matriz L
iL = inv(L);          % inversa de L
Z = @(x) iL*(x-meq); % função Z(X)
xz = @(z) L*z+meq;   % função X(Z)
dz = 1e-3;            % dz para calculo dos gradientes (diferenças finitas)
zn = [0 0 0]';        % estimativa inicial
```

```
crit = 1;
k=0;
%%%%%%%%%%%%%%%%%%%%
%% LOOP PRINCIPAL %%
%%%%%%%%%%%%%%%%%%%%
while crit>1e-8 % critério de parada
    zo = zn;
    X = xz(zo);
    % para cada variavel calcular media e desvio padrão normal equivalente
    for i=1:length(m)
        if strcmp(dists{i},'normal')
            Seq(i,i)=S(i,i);
            meq(i)=m(i);
        elseif strcmp(dists{i},'lognormal')
            [meq(i) Seq(i,i)] = eqLN(X(i),m(i),S(i,i));
        elseif strcmp(dists{i},'gumbel')
            [meq(i) Seq(i,i)] = eqT1(X(i),m(i),S(i,i));
        end
    end
    C = Seq*R*Seq;
    L = chol(C,'lower');
    iL = inv(L);
    Z = @(x) iL*(x-meq);
    xz = @(z) L*z+meq;
    % calculo do gradiente
    grad = zeros(length(m),1);
    for i=1:length(m)
        zod = zo;
        zod(i) = zo(i)-dz;
        grad(i) = ( g(xz(zo))-g(xz(zod)) )/dz;
    end
    % novo vetor de variaveis normalizadas
    zn = grad * (grad'*zo-g(xz(zo)))/(grad'*grad);
    k=k+1;
    % mudança no vetor de variaveis normalizadas
    crit = sqrt((zn-zo)'*(zn-zo));
end
mpfp.k = k;                       % número de iterações
mpfp.x = xz(zn);                  % most probable failure point
mpfp.beta = sqrt(zn'*zn);         % beta
mpfp.Pf = normcdf(-mpfp.beta); % probabilidade de falha
```

2.1 Cálculo das normais equivalentes

Funções auxiliares foram criadas para calcular médias e desvios padrão normais equivalentes de variáveis não normais. Estas funções são chamadas pela rotina form.m.

Para variáveis de distribuição lognormal:

eqLN.m

```
% compute equivalent normal mean and deviation for
% lognormal distribution
function [mueq sigmaeq] = eqLN(x,mu,s)
zeta = sqrt(log(1+(s/mu)^2));
lambda = log(mu)-zeta^2/2;
sigmaeq = normpdf(norminv(logncdf(x,lambda,zeta)))/lognpdf(x,lambda,zeta);
mueq = x-sigmaeq*norminv(logncdf(x,lambda,zeta));
```

Para variáveis de distribuição de extremos tipo I (Gumbel):

eqT1.m

```
% compute equivalent normal mean and deviation for
% type 1 extreme distribution
function [mueq sigmaeq] = eqT1(x,mu,s)
zeta = sqrt(log(1+(s/mu)^2));
lambda = log(mu)-zeta^2/2;
alpha = pi/(s*sqrt(6));
u = mu-0.5772/alpha;
t1cdf = @(x) exp(-exp(-alpha*(x-u)));
t1pdf = @(x) alpha*exp(-alpha*(x-u)-exp(-alpha*(x-u)));
sigmaeq = normpdf(norminv(t1cdf(x)))/t1pdf(x);
mueq = x-sigmaeq*norminv(t1cdf(x));
```

3 Método Monte Carlo

montecarlo.m

```
% montecarlo(g,m,s,R,n,dists)
%
% Entradas:
%   g: handle para a função de estado limite
%   m: vetor de médias
%   s: vetor de desvios padrão
%   R: matriz de correlação
%   n: número de pontos aleatórios para avaliar a f.e.l.
%   dists: nome das distribuições
% Saídas:
%   O: estrutura contendo a probabilidade de falha (Pf), o coeficiente de
%      variação (CV) e o beta (beta).
function O = montecarlo(g,m,s,R,n,dists)
S = diag(s);          % matriz S
C = S*R*S;            % matriz de covariância
L = chol(R,'lower'); % fatoração da matriz de correlação
% n conjuntos de variáveis normais não correlacionadas
Z = normrnd(0,1,[length(m) n]);
Zc = L*Z;             % imposição de correlação
Uc = normcdf(Zc);     % variáveis correlacionadas distribuidas uniformemente
% variáveis correlacionadas nas distribuições desejadas
X = zeros(size(Z));
for i=1:length(m)
    if strcmp(dists{i},'normal')
        X(i,:)= norminv(Uc(i,:),m(i),s(i));
    elseif strcmp(dists{i},'lognormal')
        zeta = sqrt(log(1+(s(i)/m(i))^2));
        lambda = log(m(i))-zeta^2/2;
        X(i,:)= logninv(Uc(i,:),lambda,zeta);
    elseif strcmp(dists{i},'gumbel')
        alpha = pi/(s(i)*sqrt(6));
        u = m(i)-0.5772/alpha;
        X(i,:)= (-log(-log(Uc(i,:)))+alpha*u)/alpha;
    end
end
Pf = sum(g(X)<0)/n;          % probabilidade de falha
O.Pf = Pf;
O.CV = sqrt((1-Pf)/(n*Pf)); % coeficiente de variação
O.beta = norminv(1-Pf);     % beta
```

4 Exemplos

Nesta seção as funções apresentadas anteriormente serão aplicadas à resolução dos exemplos 6.7 a 6.10 do livro Probability Concepts in Engineering Planning and Design (Ang & Tang, 1996).

Os exemplos a serem resolvidos têm a mesma função de estado limite:

$$g(\boldsymbol{X}) = X_1 X_2 - X_3$$

O vetor de médias e desvios padrão é o mesmo para cada exemplo:

$$\boldsymbol{M} = \{40 \;\; 50 \;\; 1000\}^T$$
$$\boldsymbol{S} = \{5 \;\; 2{,}5 \;\; 200\}^T$$

Para cada exemplo, as variáveis aleatórias assumem distribuições diferentes e/ou são correlacionadas.

4.1 Exemplo Ang & Tang 6.7

As três variáveis aleatórias seguem a distribuição normal e não são correlacionadas.

Determinação da confiabilidade pelo método FORM:

form67.m

```
% exemplo 6.7 ang
% método FORM
g = @(x) x(1)*x(2)-x(3);              % função de estado limite
M = [40 50 1000]';                    % vetor de médias
CV = [0.125 0.05 0.2]';               % vetor de coeficientes de variação
S = M.*CV;                            % vetor de desvios padrão
r = eye(3);                           % matriz de correlação
dists = {'normal' 'normal' 'normal'}; % distribuições
mpfp = form(g,M,S,r,dists)            % resultados
```

Resultado:

```
> form67
mpfp =

  scalar structure containing the fields:

    k =  10
    x =

         28.550
         48.308
       1379.219

    beta =  3.0491
    Pf =  0.0011477
```

Determinação da confiabilidade pelo método MC:

mc67.m

```
% exemplo 6.7 ang
% função de estado limite
g = @(x) x(1,:).*x(2,:)-x(3,:);
M = [40 50 1000]';
CV = [0.125 0.05 0.2]';
S = M.*CV;
r = eye(3);
dists = {'normal' 'normal' 'normal'};
n = 4e6;
O = montecarlo(g,M,S,r,n,dists)
```

Resultado:

```
> mc67
O =

  scalar structure containing the fields:

    Pf =  0.0011620
    CV =  0.014659
    beta =  3.0454
```

4.2 Exemplo Ang & Tang 6.8

As variáveis aleatórias X_1 e X_2 seguem distribuições lognormais e X_3 segue a distribuição de Gumbel. As variáveis não são correlacionadas.

Determinação da confiabilidade pelo método FORM:

form68.m

```
% exemplo 6.8 ang
g = @(x) x(1)*x(2)-x(3);                          % função de estado limite
M = [40 50 1000]';                                % vetor de médias
CV = [0.125 0.05 0.2]';                           % vetor de coeficientes de variação
S = M.*CV;                                        % vetor de desvios padrão
r = eye(3);                                       % matriz de correlação
dists = {'lognormal' 'lognormal' 'gumbel'};       % distribuições
mpfp = form(g,M,S,r,dists)                        % resultados
```

Resultado:

```
> form68
mpfp =

  scalar structure containing the fields:

    k =  9
    x =

         34.299
         48.777
       1673.030

    beta =  2.7422
    Pf =  0.0030511
```

Determinação da confiabilidade pelo método MC:

mc68.m

```
% exemplo 6.8 ang
% função de estado limite
g = @(x) x(1,:).*x(2,:)-x(3,:);
M = [40 50 1000]';
CV = [0.125 0.05 0.2]';
S = M.*CV;
r = eye(3);
dists = {'lognormal' 'lognormal' 'gumbel'};
n = 4e6;
O = montecarlo(g,M,S,r,n,dists)
```

Resultado:

```
> mc68
O =

  scalar structure containing the fields:

    Pf =  0.0031110
    CV =  0.0089504
    beta =  2.7358
```

4.3 Exemplo Ang & Tang 6.9

As três variáveis aleatórias seguem a distribuição normal. As variáveis X_1 e X_2 são correlacionadas com ρ=0,4.

Determinação da confiabilidade pelo método FORM:

form69.m

```
% exemplo 6.9 ang
% metodo form
g = @(x) x(1)*x(2)-x(3);            % função de estado limite
M = [40 50 1000]';                  % vetor de médias
CV = [0.125 0.05 0.2]';             % vetor de coeficientes de variação
S = M.*CV;                          % vetor de desvios padrão
r = [1 .4 0; .4 1 0; 0 0 1];        % matriz de correlação
dists = {'normal' 'normal' 'normal'}; % distribuições
mpfp = form(g,M,S,r,dists)          % resultados
```

Resultado:

```
> form69
mpfp =

  scalar structure containing the fields:

    k =  8
    x =

         28.857
         46.479
       1341.233

    beta =  2.8629
    Pf =  0.0020991
```

Determinação da confiabilidade pelo método MC:

mc69.m

```
% exemplo 6.9 ang
% função de estado limite
g = @(x) x(1,:).*x(2,:)-x(3,:);
M = [40 50 1000]';
CV = [0.125 0.05 0.2]';
S = M.*CV;
r = [1 .4 0; .4 1 0; 0 0 1];
dists = {'normal' 'normal' 'normal'};
n = 4e6;
O = montecarlo(g,M,S,r,n,dists)
```

Resultado:

```
> mc69
O =

  scalar structure containing the fields:

    Pf =  0.0021215
    CV =  0.010844
    beta =  2.8595
```

4.4 Exemplo Ang & Tang 6.10

As variáveis aleatórias X_1 e X_2 seguem distribuições lognormais e X_3 segue a distribuição de Gumbel. As variáveis X_1 e X_2 são correlacionadas com $\rho=0,4$.

Determinação da confiabilidade pelo método FORM:

form610.m

```
% exemplo 6.10 ang
% método FORM
g = @(x) x(1,:).*x(2,:)-x(3,:);          % função de estado limite
M = [40 50 1000]';                        % vetor de médias
CV = [0.125 0.05 0.2]';                   % vetor de coeficientes de variação
S = M.*CV;                                % vetor de desvios padrão
r = [1 .4 0; .4 1 0; 0 0 1];              % matriz de correlação
dists = {'lognormal' 'lognormal' 'gumbel'}; % distribuições
mpfp = form(g,M,S,r,dists)                % resultados
```

Resultado:

```
> form610
mpfp =

  scalar structure containing the fields:

    k =  9
    x =

         33.785
         47.757
       1613.484

    beta =  2.6646
    Pf =  0.0038538
```

Determinação da confiabilidade pelo método MC:

mc610.m

```
% exemplo 6.10 ang
% função de estado limite
g = @(x) x(1,:).*x(2,:)-x(3,:);
M = [40 50 1000]';
CV = [0.125 0.05 0.2]';
S = M.*CV;
r = [1 .4 0; .4 1 0; 0 0 1];
dists = {'lognormal' 'lognormal' 'gumbel'};
n = 4e6;
O = montecarlo(g,M,S,r,n,dists)
```

Resultado:

```
> mc610
O =

  scalar structure containing the fields:

    Pf =  0.0039610
    CV =  0.0079288
    beta =  2.6554
```