**Arch Bridge Optimization under Uniform Pressure Considering Fracture Stress**

2017-82260 Yann De Coster

2017-20541 Jin Uk Ko

1. **Abstract**

In the class “*Probabilistic Engineering Analysis and Design*” led by professor Youn, three important issues encountered in practical engineering fields were covered: Reliability assessment, Reliability-based design, and Reliability management. Thanks to his instruction, we could handle a practical optimization of design variables using Design Optimization (DO) and Reliability-Based Design Optimization (RBDO) for a uniform distributed load on an arch bridge. We formulate the constrained optimization problem defined for this problem and perform finite element method (FEM) with COMSOL. This report mainly consists of four different parts: Problem statement, Reliability analysis, Reliability analysis for initial design and DO, and Reliability-Based Design Optimization.

At first, a description and a motivation of this project problem is given. Then a Monte Carlo Simulation, a First-order Reliability Method, a Second-order Reliability Method, and a Reliability Index Approach are performed to quantitatively evaluate the reliability of the initial design and DO.

1. **Nomenclature**

**x1: End width [m]**

**x2: Radius of semi-circle [m]**

**x3: Height of rectangle tunnel [m]**

**x4: Height of bridge [m]**

**E: Young`s modulus [GPa]**

**: Density [kg/m^3]**

**: Poisson ratio**

**: Fracture stress (compressive strength)**

1. **Introduction**

In Europe Arch bridges represent nearly 80% of architectural monuments which have a reach above 2 meters. Most of the arch bridges are done with brick masonry, which is a quite cheap material. And this material and technology are proven resistant as they have stood the test of time. Back in time, more than 2 millennium ago, Romans had created aqueducts and bridges which still survive and be as strong as when they were built [1].

Although those impressive monuments seem almost invulnerable, it happens that they collapse, resulting in a high cost of money, and even sometimes leading to deaths and injuries [2]. In Tours, during the spring of 1978, an arch of the Wilson Bridge was collapsing while a car was driving across [3] it. Luckily, during this one-hundred-thousand inhabitant French city breakdown, the driver had the reflex to speed his car up, thus fleeing his possible death. Even though everyone was safe, the city had to rebuild the bridge as it is part of the city heritage. The option chosen was to consolidate the preserved parts of the bridge and to reconstruct the collapsed ones. This option had a cost of 35 million of francs: the equivalent of today 18.4 million euros. In addition of this very high cost, the bridge was unusable for 18 months, loading up the city road network.

Sometimes, the consequences of a bridge collapse can even be worse. For example, in December 2015, in the British city of Tadcaster, the Wharfe Bridge breakdown had caused the fracture of the main city gas pipeline, incurring the evacuation of hundreds of residents [4]. The bridge repair took 13 months with a cost of 4.4 million pounds, and nearby inhabitants required a long detour for crossing the river by car. Another disastrous event is the collapse of Xiushui County in August 2016, which led to three deaths and two injuries [5].

All those tragedies were caused by previous floods and by current running water scouring. The purpose of this project is to reduce the number of failures while reducing the construction cost due to the volume of material used.

1. **Problem statement**

Information of random variables for our problem is tabulated below.

Table 1. Information of random variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Random Variable** | **Mean** | **Standard Deviation** | **Distribution Type** | **Lower Design Bound** | **Upper Design Bound** |
| **x1 [m]** | 2 | 0.3 | normal | 4 | 10 |
| **x2 [m]** | 8.5 | 0.6 | normal | 10 | 20 |
| **x3 [m]** | 2.5 | 0.1 | normal | 1 | 4 |
| **x4 [m]** | 17 | 0.8 | normal | 24 | 36 |
| **E [GPa]** | 20 | 1.0 | normal | 10 | 30 |
| **[kg/m^3]** | 2000 | 20 | normal | 1800 | 2200 |
| **P [kN/m]** | 500 | 25 | normal | 400 | 600 |
|  | Deterministic: 0.15 | | | | |
| **Length [m]** | Deterministic: 100 | | | | |

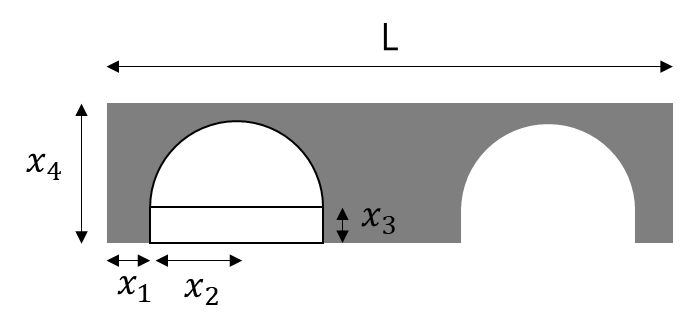
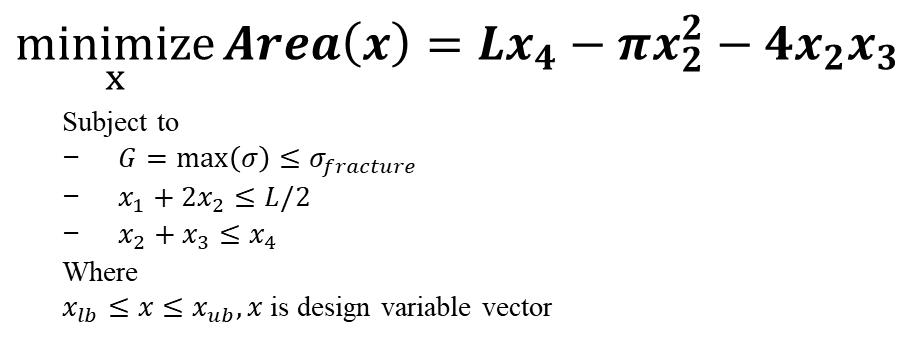
****

Figure 1. Shape of arch bridge

The goal of our RBDO is to design vector of the arch bridge while satisfying given reliability condition. Fig. 1 is the shape of arch bridge. We assume this problem is two dimensional. L is the length of bridge, and the arch consists of a semi-circle & a rectangle.

Now we will explain how to choose to mean, standard deviation, bounds, and distributions of variables. Firstly, we assume the length of bridge is deterministic, because the dimension of place where a bridge will be laid is determined. Then we let x1~x4 have a right mean considering length, and thickness. According to a study about properties of brick masonry, young`s modulus, Poisson ratio, and density are between 7-34[GPa], 0.1-0.3, and 1800-2200[kg/m^3] [6], respectively. So, we set the means of those three variables as 20[GPa], 2000[kg/m^3], and 0.15. Among these variables, Poisson ratio usually considered deterministic [7], so we let it deterministic. Assuming the loading of bridge multiple truck loading [8], we suppose 5-ton trucks fill the bridge completely. Then the distributed load is about 500 [kN/m]. Finally, we assume all random variables follow normal distributions for simplicity.

We formulate design optimization problem like below. The optimization problem is non-linear constrained problem.



Cost of brick is roughly between $6.50 and $10.00 per square foot [9]. We set objective function as the area of bridge (i.e. area of Fig. 1), because we want to minimize construction cost. The first constraint means maximum Von-mises stress of bridge should be less than compressive strength (. If maximum stress exceeds compressive strength, fracture can occur [10]. We set as 3[MPa] with reference to a study mentioned above [6]. Stress is calculated by CAE program (COMSOL). The third and fourth ones are for geometry. The former one is to prevent from exceeding half of length, and the latter one is to prevent from bigger than . These two conditions are linear inequality conditions. In RBDO, we will give probability constraints on the first conditions.

Before solving the problem, we check the convergence of CAE model with respect to mesh size. The result is shown in Fig. 2. We use auto-mesh, because of circular shape. The bigger mesh size in Fig. 2, the finer mesh size. Considering convergence and computational cost, we choose the second finest mesh size.

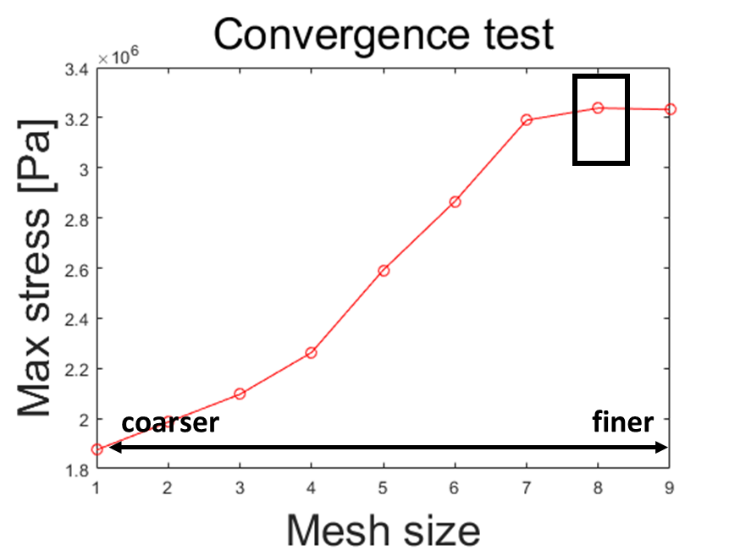


Figure 2. Convergence test of model

1. **Reliability Analysis**
   1. Reliability analysis for the initial design

We will use (i) Monte Carlo Simulation (MCS), (ii) First-order Reliability Method (FORM), (iii) Second-order Reliability Method (SORM), and (iv) Reliability Index Approach (RIA). In all method, target reliability is 0.99. That is, if calculated reliability is smaller than 0.99, it is considered failure.

1. Monte Carlo Simulations (MCS)

In MCS, we simulate n samples, and the ones satisfying are regarded reliable samples. Then reliability about is calculated like below.

1. First-order Reliability Method (FORM)

In FORM, we perform Taylor expansion about mean of x, and approximate by first-order. Then mean of , and variance of are like below.

Then reliability about is calculated like below.

1. Second-order Reliability Method (SORM)

SORM is similar with FORM, but the difference is SORM approximates by second-order. Mean of , and variance of are like below.

Then reliability about is calculated like below.

1. Reliability Index Approach (RIA)

In RIA, most-probable-point (MPP) of is a point on having the shortest distance from origin. We find MPP of by HL-RF method, and calculate reliability about like below.

We can also calculate sensitivity in RIA. Sensitivity means the variation of when changes. By analyzing sensitivity, we can find the behavior of with respect to

When calculating first-order gradient, we use numerical method. Especially, we employ FD1 method for higher accuracy. We calculate the difference of , and then divide it by . is determined by .

* 1. Reliability analysis for the initial design

The initial design is like below. As mentioned before, is a design vector.

Table 2. Initial design

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x1(m)** | **x2(m)** | **x3(m)** | **x4(m)** | **E(GPa)** |  | **P(kPa)** |  | **L(m)** |
| 7 | 15 | 2.5 | 30 | 20 | 2000 | 500 | 0.15 | 100 |

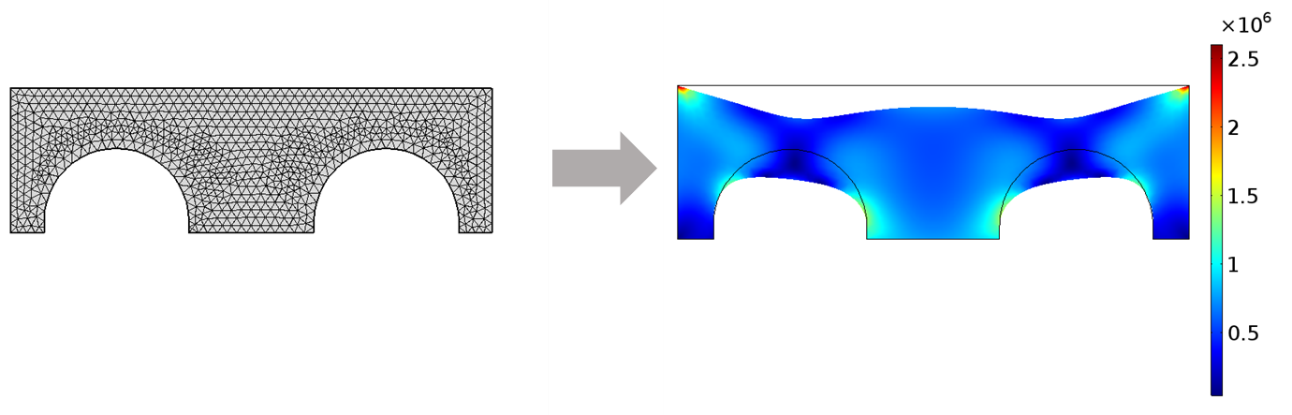


Figure 3. Model of initial design

We perform the reliability analysis for the initial design in each method mentioned in 2.1. The results are shown below.

Table 3. Reliability for initial design with several methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **MCS** | **FORM** | **SORM** | **RIA** |
| **Maximum stress** | 0.4387 | 0.4838 | 0.4839 | 0.4812 |

All methods show initial design is not reliable. MCS shows a little bit different result with the rest methods. We guess the reason is the short of samples (We perform 1500 samples in MCS for short of time)

Fig. 4 is the pdf and cdf in MCS, FORM, and SORM. We can find FORM and SORM almost similar, but they are a little different from MCS. If we increase the number of samples, the difference might be smaller.

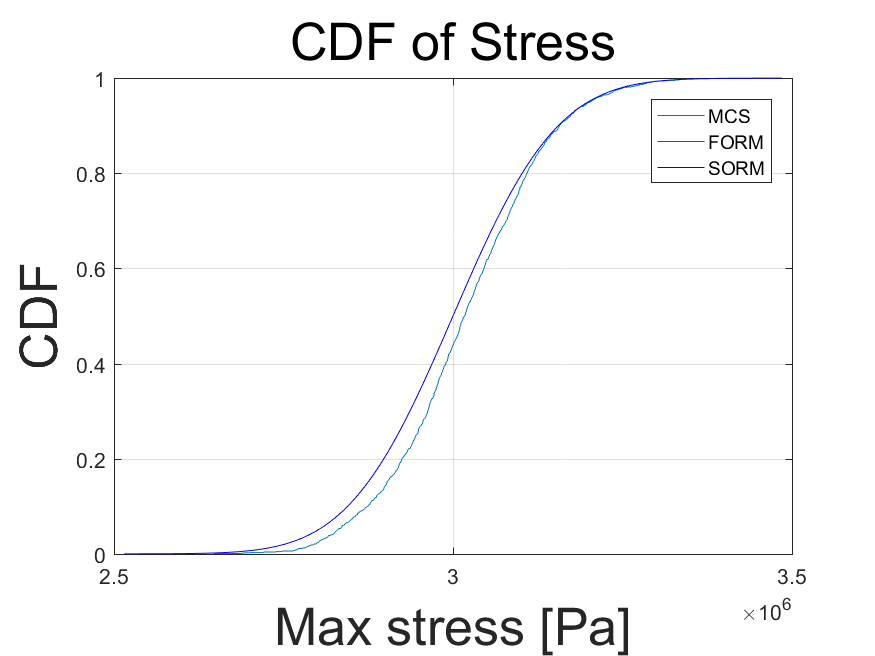
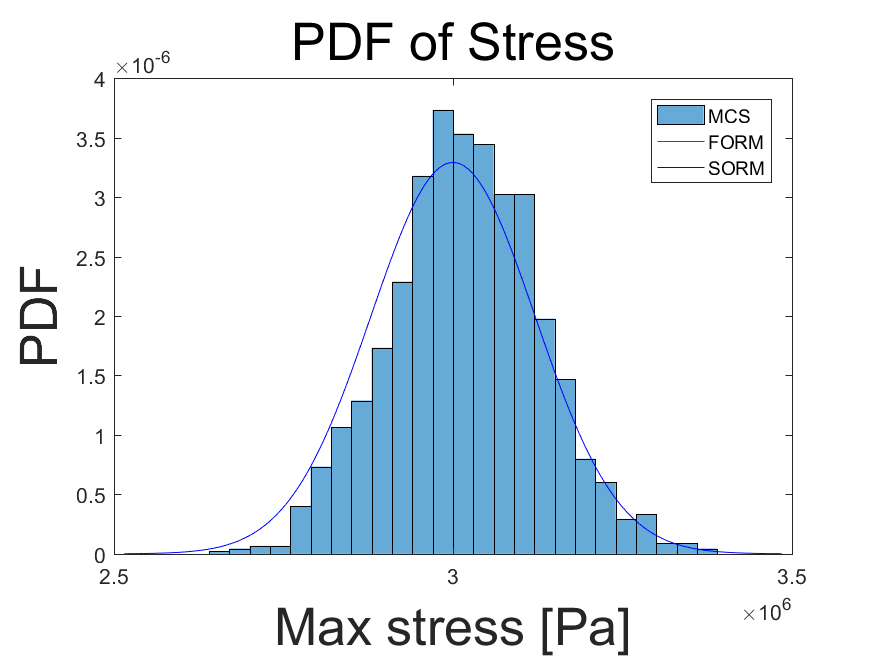
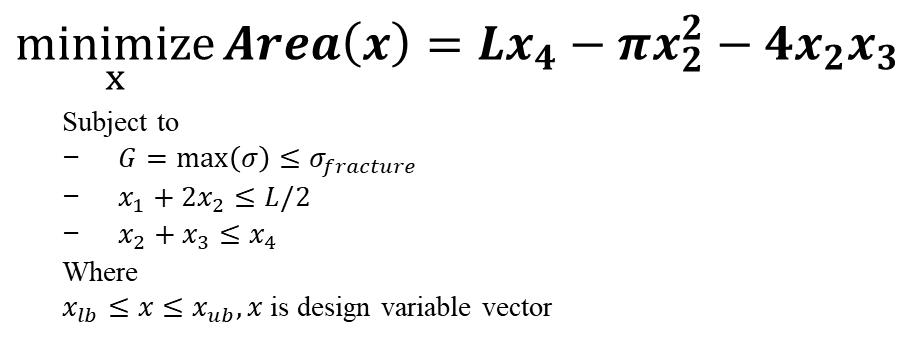


Figure 4. PDF & CDF of stress

1. **Reliability-Based Design Optimization**
   1. Design Optimization (DO)

The mathematical statement of design optimization is like below. We do not consider reliability.



We set initial design as mentioned in part 2, and solve optimization problem by MATLAB. The optimal design is like below.

Table 4. Result of DO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **x1(m)** | **x2(m)** | **x3(m)** | **x4(m)** |
| **Initial design** | 7 | 15 | 2.5 | 30 |
| **Optimal design** | 7 | 13.3403 | 2.0021 | 24 |

|  |  |
| --- | --- |
| C:\Users\user\Desktop\RBDO\2D_171206\2D__initial_stress.png | C:\Users\user\Desktop\RBDO\2D_171206\2D_DDO__stress.png |

Figure 5. Initial & Optimal design model

Looking at the result, firstly we can find that x1 does not change. It should be x1 does not affect objective function directly. As a matter of fact, we knew the fact area does not depend on x1, but we thought x1 can affect stress and displacement conditions. Hence, we leave it as a design variable. Secondly, design variables x2 & x3 decrease. We guess these two variables are quite related to stress condition. Lastly, x4 decreases to lower bound. This means x4 does not heavily affect stress.

* 1. Reliability analysis for Design Optimization (DO)

We perform reliability analysis for DO in MCS, FORM, SORM, and RIA. The results are shown below.

Table 5. Reliability for DO with several methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **MCS** | **FORM** | **SORM** | **RIA** |
| **Maximum stress** | 0.4610 | 0.5000 | 0.5001 | 0.5000 |

FORM, SORM, and RIA show similar reliabilities but MCS shows a little bit different result. This might be because the number of sample is small (We perform 1500 samples for short of time). We plot the PDF & CDF of stress in Fig. 6. As can be seen in that plot, FORM and SORM are almost same but different from MCS.

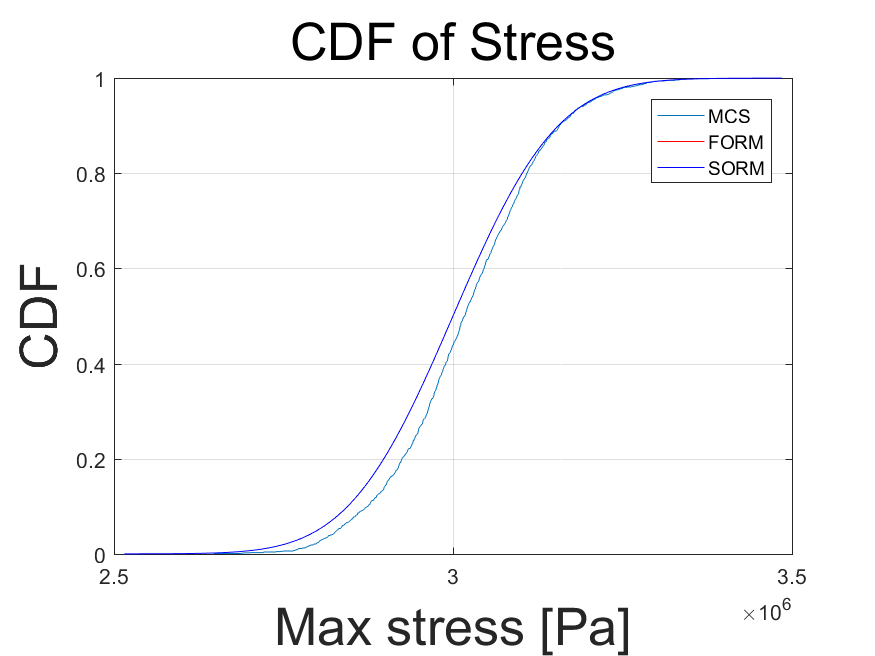
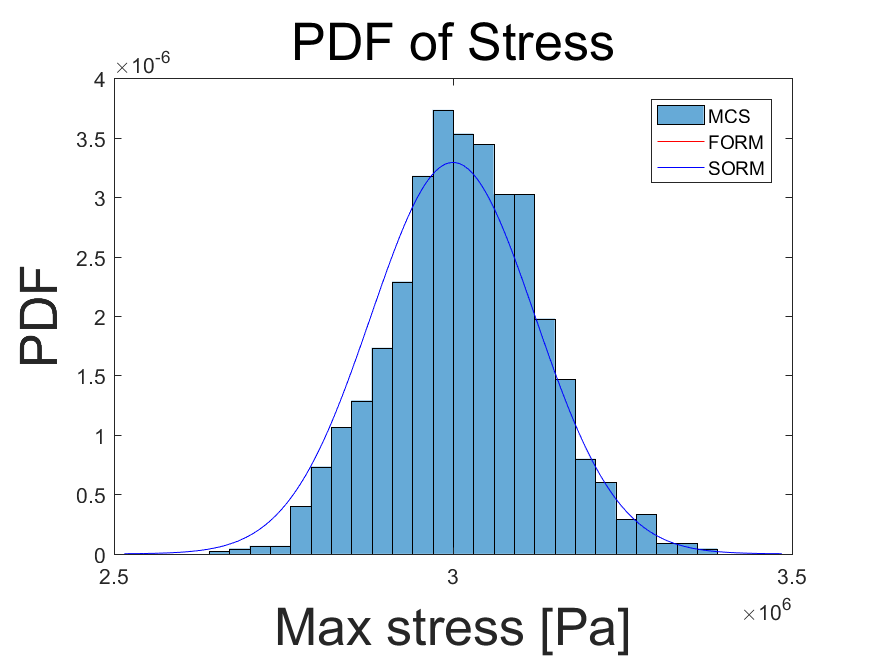
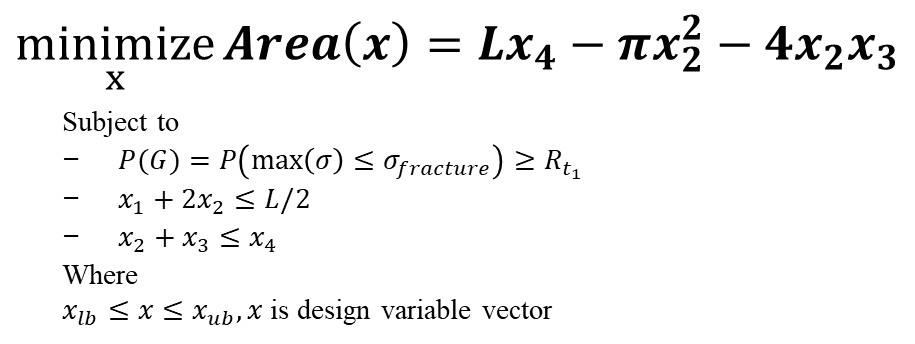


Figure 6. PDF & CDF of stress

* 1. Reliability-Based Design Optimization (RBDO)

We formulate RBDO problem like below. We give probability condition on the first and second constraints. The target reliability is 0.99.



We wanted to perform RBDO 4 times─starting point (Initial vs DO) & Algorithm (PMA vs RIA). Unfortunately, the computational time of our RBDO code was much bigger than expected: it was longer than a day. We thought there might be two reasons. First reason is problem in convergence of our problem due to non-convexity. Second is the curved shape (semi-circle) of our model makes CAE analysis longer than normal problem. Frankly speaking, we initially made our problem 3D like below figure. But it took too much time to run RBDO code. Hence, we decided to remove one dimension in our initial problem, and performed the project so far. Sadly, the computational time was still too high for us to have proper results, which explains why we removed the RBDO part. However, the RBDO code remains available in the code section of this paper.

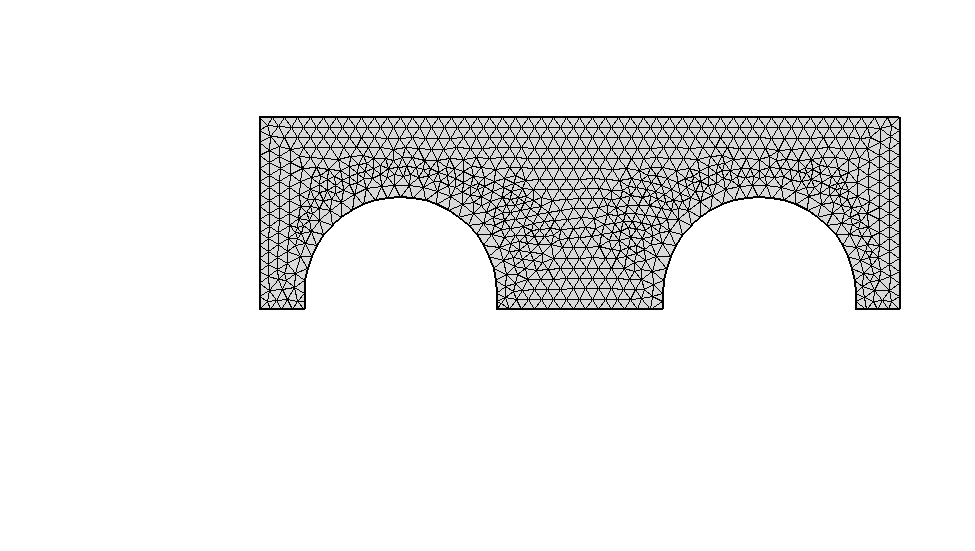
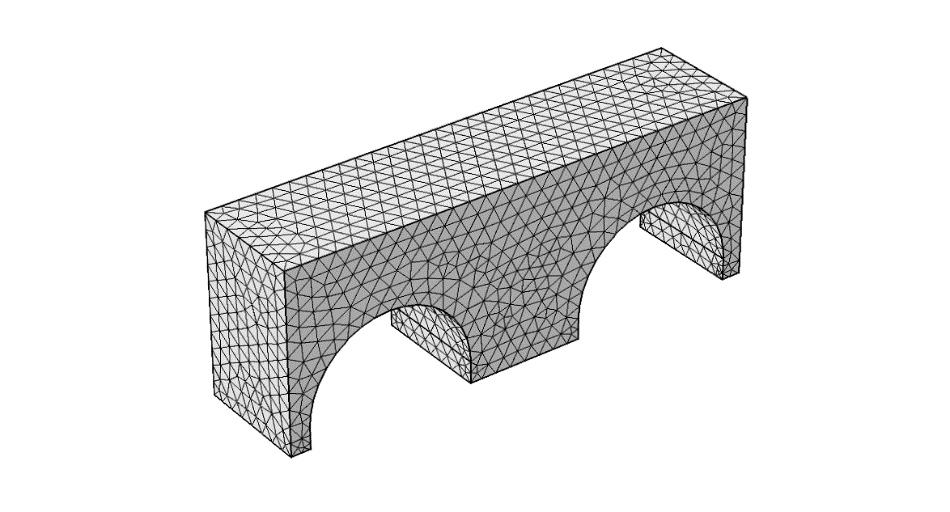


Figure 7. 3D model vs 2D model

1. **References**

[1] Jean-Michel Delbecq, Les ponts en maçonnerie, Ministère des Transports, 1982

[2] Load distribution on arch bridges, http://ginahagler.com/forces-at-work-arch-bridges/

[3] Wilson Bridge, https://fr.wikipedia.org/wiki/Pont\_Wilson\_(Tours)

http://archives.cg37.fr/UploadFile/GED/Actualites/1207676521.pdf

[4] Wharfe Bridge, https://en.wikipedia.org/wiki/Tadcaster\_Bridge, http://www.dailymail.co.uk/news/article-3378677/Town-cut-two-130-people-return-homes-collapse-300-year-old-bridge-floods.html

[5] iushui County Bridge, https://thenanfang.com/bridge-collapse-kills-3-east-china/

http://news.xinhuanet.com/english/2016-08/22/c\_135622764\_3.htm

[6] Narayanan S P, Sirajuddin M, “Properties of brick masonry for FE modeling”, AJER, 2013, vol. 1, pp. 06-11.

[7] Daniel V. Oliveira, “Mechanical characterization of stone and brick masonry”, Graduate thesis, Apr. 2000.

[8] Bridge Load Evaluation Manual, Dec.2016, Alberta Gov.

[9] “Cost of brick masonry”, http://www.riversidebrick.com/products/pricing-guide/, Dec. 2017.

[10] “Compressive strength”, https://en.wikipedia.org/wiki/Compressive\_strength, Dec. 2017.

1. **Code**

|  |
| --- |
| %% Reliability Based Design Optimization  function Arch\_bridge\_RBDO()  %% Material Properties  clc; close all; clear all;  global nc nr nm ub lb bt stddev iter stress\_allow    nc = 1; % Number of contraints  nr = 7; % Number of variable  nm = 1; % Number of mode, nm=1: AMV, nm=2: RIA  bt = norminv(0.99,0,1); % Target Reliability    %% Initial point  x0 = [7, 15, 2.5, 30, 20, 2000, 500]; % 2D initial design  % x0 = [7 13.340319 2.002096 24 20 2000 500]; % 2D DO  stddev = [0.3, 0.6, 0.1, 0.8, 1.0, 100, 25];    %% Upper/Lower Bound  ub = [10, 20, 4, 36, 30, 2200, 600]; % Upper Bound  lb = [4, 10, 1, 24, 10, 1800, 400]; % Lower Bound    poisson = 0.15;  length = 100;    stress\_allow = 3.0e6;    %% Linear inequailty Constraint  A = [1 2 0 0 0 0 0; 0 1 1 -1 0 0 0];  b = [length/2; 0];    %% Reliability based Optimization  iter = 0;  x\_start = x0;  options = optimset('LargeScale', 'off', 'DiffMaxChange', 0.1,'DiffMinChange', 0.001,...  'TolX', 1E-5, 'Tolcon', 1E-5, 'Tolfun', 1E-5, 'GradObj', 'on', 'GradConstr', 'off', ...  'Algorithm', 'sqp', 'Display', 'iter');  [x\_optm, fval, exit\_flag]= fmincon(@CostFun,x0,A,b,[],[],lb,ub,@ReliConst,options)    %% Cost Function  function [f, gradf] = CostFun(x)  f = x(4)\*length-pi\*x(2)^2-4\*x(2)\*x(3);  gradf = [0, -2\*pi\*x(2)-4\*x(3), -4\*x(2), length, 0, 0, 0];  end    %% Transfer to U-space  function [X,dXdU] = x2u(u, x)  X = x+u.\*stddev;  dXdU = stddev;  end    %% Constraints and Gradients  function [c, ceq, GC, GCeq] = ReliConst(x)  ceq = []; GCeq = [];  for kc = 1:nc  if nm==1  [G,dG] = AMV(x,kc);  beta(kc) = G;  dbeta(:, kc) = dG./stddev;  elseif nm==2  [G,dG] = HL\_RF(x,kc);  beta(kc) = bt-G;  dbeta(:,kc) = -dG;  end  end  c = beta;  GC = dbeta;  dx = norm(x-x\_start);  x\_start = x;  end    %% PMA  function [G, dGdX] = AMV(x,kc)  u = zeros(1,nr); Dif=1; iter=0;  while Dif>=1d-5 && iter<20  iter=iter+1;  if iter>1  u = dGdX\*bt/norm(dGdX);  end  [G, dGdX] = Constraints(u,x,kc);  U(iter,:)=u/bt;  if iter>1  Dif=abs(U(iter,:)\*U(iter-1,:)'-1);  end  end  end    %% RIA  function [beta, dbeta] = HL\_RF(x0,kc)  u = zeros(1,nr); Dif=1; iter=0;  while Dif>=1d-5 && iter<20  iter = iter+1;  [G, dGdX] = Constraints(u,x0,kc);  n = dGdX/norm(dGdX);  u = (u\*n'-G/norm(dGdX))\*n;  U(iter,:) = u/norm(u);  if iter==1  sign = -G/abs(G);  elseif iter>1  Dif = abs(U(iter,:)\*U(iter-1,:)'-1);  end  end  beta = norm(u)\*sign;  [~, dxdu] = x2u(u,x0);  dbeta = -u./(beta.\*dxdu);  Reliability = normcdf(beta);  end    %% Constraints  function [G, dGdX] = Constraints(u, x, kc)  [X, dXdU] = x2u(u, x);  max\_von\_mises = Arch\_bridge\_2D([X poisson length]);  const = max\_von\_mises-stress\_allow;  G = const(kc);    for i=1:nr  delta\_x = zeros(1,nr);  delta\_x(i) = 1e-4\*x(i);  x\_plus\_delta\_x = x + delta\_x;  [X\_plus\_delta, ~] = x2u(u, x\_plus\_delta\_x);  max\_von\_mises\_plus\_delta\_x = Arch\_bridge\_2D([X\_plus\_delta poisson length]);  const\_plus\_delta\_x = max\_von\_mises\_plus\_delta\_x - stress\_allow;  G\_plus\_delta\_x = const\_plus\_delta\_x(kc);  dGdX(i)=(G\_plus\_delta\_x-G)/delta\_x(i);  end  dGdX = dGdX.\*dXdU;  end    end |
| function [max\_von\_mises] = Arch\_bridge\_2D(x)  %  % arch\_bridge\_2D\_after\_study.m  %  % Model exported on Dec 6 2017, 15:35 by COMSOL 5.1.0.234.    import com.comsol.model.\*  import com.comsol.model.util.\*    model = ModelUtil.create('Model');    model.modelPath('C:\Users\SHRM\Desktop\Jin Uk\RBDO\_171206');    model.comments(['Untitled\n\n']);    model.modelNode.create('comp1');    model.geom.create('geom1', 2);    model.mesh.create('mesh1', 'geom1');    model.physics.create('solid', 'SolidMechanics', 'geom1');    model.study.create('std1');  model.study('std1').create('stat', 'Stationary');  model.study('std1').feature('stat').activate('solid', true);    model.param.set('x1', [num2str(x(1)) '[m]']);  model.param.set('x2', [num2str(x(2)) '[m]']);  model.param.set('x3', [num2str(x(3)) '[m]']);  model.param.set('x4', [num2str(x(4)) '[m]']);  model.param.set('young', [num2str(x(5)) '[GPa]']);  model.param.set('density', [num2str(x(6)) '[kg/m^3]']);  model.param.set('pressure', [num2str(x(7)) '[kN/m]']);  model.param.set('poisson', num2str(x(8)));  model.param.set('length', [num2str(x(9)) '[m]']);    model.geom('geom1').run('');  model.geom('geom1').feature.create('r1', 'Rectangle');  model.geom('geom1').feature('r1').set('type', 'solid');  model.geom('geom1').feature('r1').set('base', 'corner');  model.geom('geom1').feature('r1').set('pos', {'-1.3' '-0.05'});  model.geom('geom1').feature('r1').set('size', {'1' '0.35'});  model.geom('geom1').run('r1');  model.geom('geom1').feature('r1').set('size', {'length' 'x4'});  model.geom('geom1').feature('r1').set('pos', {'0' '0'});  model.geom('geom1').run('r1');  model.geom('geom1').run('r1');  model.geom('geom1').feature.create('r2', 'Rectangle');  model.geom('geom1').feature('r2').set('type', 'solid');  model.geom('geom1').feature('r2').set('base', 'center');  model.geom('geom1').feature('r2').set('pos', {'8' '4'});  model.geom('geom1').feature('r2').set('size', {'20' '8'});  model.geom('geom1').run('r2');  model.geom('geom1').feature('r2').set('size', {'2\*x2' 'x3'});  model.geom('geom1').feature('r2').set('base', 'corner');  model.geom('geom1').feature('r2').set('pos', {'x1' '0'});  model.geom('geom1').run('r2');  model.geom('geom1').run('r2');  model.geom('geom1').feature.create('c1', 'Circle');  model.geom('geom1').feature('c1').set('type', 'solid');  model.geom('geom1').feature('c1').set('base', 'corner');  model.geom('geom1').feature('c1').set('pos', {'4' '0'});  model.geom('geom1').feature('c1').set('r', '4');  model.geom('geom1').run('c1');  model.geom('geom1').feature('c1').set('r', 'x2');  model.geom('geom1').feature('c1').set('angle', '180');  model.geom('geom1').feature('c1').set('base', 'center');  model.geom('geom1').feature('c1').set('pos', {'x1+x2' 'x3'});  model.geom('geom1').run('c1');  model.geom('geom1').run('c1');  model.geom('geom1').create('mir1', 'Mirror');  model.geom('geom1').feature('mir1').selection('input').set({'c1' 'r2'});  model.geom('geom1').feature('mir1').set('keep', 'on');  model.geom('geom1').feature('mir1').set('pos', {'length/2' '0'});  model.geom('geom1').run('mir1');  model.geom('geom1').run('mir1');  model.geom('geom1').create('dif1', 'Difference');  model.geom('geom1').feature('dif1').selection('input').set({'r1'});  model.geom('geom1').feature('dif1').selection('input2').set({'c1' 'mir1' 'r2'});  model.geom('geom1').run('dif1');    model.physics('solid').feature.create('fix1', 'Fixed', 1);    model.geom('geom1').run;    model.physics('solid').feature('fix1').selection.set([1 2 6 9 10]);  model.physics('solid').feature.create('bndl1', 'BoundaryLoad', 1);  model.physics('solid').feature('bndl1').selection.set([3]);  model.physics('solid').feature('bndl1').set('FperArea', {'0' '-pressure' '0'});    model.material.create('mat1', 'Common', 'comp1');  model.material('mat1').propertyGroup('def').set('youngsmodulus', {'young'});  model.material('mat1').propertyGroup('def').set('poissonsratio', {'poisson'});  model.material('mat1').propertyGroup('def').set('density', {'density'});    model.mesh('mesh1').autoMeshSize(2);  model.mesh('mesh1').run;    model.sol.create('sol1');  model.sol('sol1').study('std1');    model.study('std1').feature('stat').set('notlistsolnum', 1);  model.study('std1').feature('stat').set('notsolnum', '1');  model.study('std1').feature('stat').set('listsolnum', 1);  model.study('std1').feature('stat').set('solnum', '1');    model.sol('sol1').create('st1', 'StudyStep');  model.sol('sol1').feature('st1').set('study', 'std1');  model.sol('sol1').feature('st1').set('studystep', 'stat');  model.sol('sol1').create('v1', 'Variables');  model.sol('sol1').feature('v1').set('control', 'stat');  model.sol('sol1').create('s1', 'Stationary');  model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');  model.sol('sol1').feature('s1').feature('fc1').set('termonres', 'auto');  model.sol('sol1').feature('s1').feature('fc1').set('reserrfact', 1000);  model.sol('sol1').feature('s1').feature('fc1').set('linsolver', 'dDef');  model.sol('sol1').feature('s1').feature('fc1').set('termonres', 'auto');  model.sol('sol1').feature('s1').feature('fc1').set('reserrfact', 1000);  model.sol('sol1').feature('s1').feature.remove('fcDef');  model.sol('sol1').attach('std1');    model.result.create('pg1', 2);  model.result('pg1').set('data', 'dset1');  model.result('pg1').create('surf1', 'Surface');  model.result('pg1').feature('surf1').set('expr', {'solid.mises'});  model.result('pg1').label('Stress (solid)');  model.result('pg1').feature('surf1').create('def', 'Deform');  model.result('pg1').feature('surf1').feature('def').set('expr', {'u' 'v'});  model.result('pg1').feature('surf1').feature('def').set('descr', 'Displacement field (Material)');    model.sol('sol1').runAll;    model.result('pg1').run;  model.result.numerical.create('max2', 'MaxSurface');  model.result.numerical('max2').selection.set([1]);  model.result.numerical('max2').set('expr', 'solid.mises');  model.result.table.create('tbl2', 'Table');  model.result.table('tbl2').comments('Surface Maximum 2 (solid.mises)');  model.result.numerical('max2').set('table', 'tbl2');  model.result.numerical('max2').setResult;    %% Von mises stress  von\_mises\_cell = struct2cell(mphtable(model, 'tbl2'));  max\_von\_mises = von\_mises\_cell{3};    out = model;  clearvars -except max\_von\_mises |
| function Arch\_bridge\_DDO\_2D()  clc; clear all; close all  %% Initial value  % nc = 10; nd = 4;  young = 20;  density = 2000;  pressure = 500;    poisson = 0.15;  length = 100;    stress\_allow = 3.0e6;    %% Upper/lower bound  ub = [10, 20, 4, 36];  lb = [4, 10, 1, 24];    x1 = mean([lb(1), ub(1)]);  x2 = mean([lb(2), ub(2)]);  x3 = mean([lb(3), ub(3)]);  x4 = mean([lb(4), ub(4)]);  x0 = [x1, x2, x3, x4]; % Initial design vector  %% Linear inequailty Constraint  A = [1 2 0 0; 0 1 1 -1];  b = [length/2; 0];    %% Optimization  Iter = 0;  options = optimoptions(@fmincon, 'GradObj', 'on', ...  'Display', 'iter', 'Algorithm', 'sqp', 'PlotFcn', @optimplotfval, ...  'TolX', 1e-9, 'TolF', 1e-9', 'TolCon', 1e-9, ...  'MaxFunEvals', 200);  % options = optimoptions(@fmincon, 'Algorithm', 'sqp');  % options = optimoptions(options, 'SpecifyObjectiveGradient', true);    tic  x\_optm = fmincon(@CostFun,x0,A,b,[],[],lb,ub,@Constraints, options);  toc    % x\_optm = fmincon(@CostFun,x0,A,b,[],[],lb,ub, [], options);  disp(x\_optm)  %==========================================================================  % Built-in function (1) Cost function  %==========================================================================  function [f, gradf] = CostFun(x)  f = x(4)\*length-pi\*x(2)^2-4\*x(2)\*x(3);  gradf = [0, -2\*pi\*x(2)-4\*x(3), -4\*x(2), length];    end  %==========================================================================  % Built-in function (2) Constraints  %==========================================================================  function [c, ceq] = Constraints(x)  xx = [x young density pressure poisson length];  max\_von\_mises = Arch\_bridge\_2D(xx);  Iter = Iter+1;  c2 = max\_von\_mises - stress\_allow;  fprintf('Iter: %d, Cost: %f, x(1): %f, x(2): %f, x(3): %f, x(4): %f\n', Iter, CostFun(x), x(1), x(2), x(3), x(4));  fprintf('Max mises: %g, c2: %g. \n', max\_von\_mises, c2);  c = c2;  ceq = [];  end    end |
| %% Monte Carlo Simulation  clc; clear all; close all;    %% Values  x0 = [7, 15, 2.5, 30, 20, 2000, 500]; % 2D initial design  % x0 = [7 13.340319 2.002096 24 20 2000 500]; % 2D DO  stddev = [0.3, 0.6, 0.1, 0.8, 1.0, 100, 25];    stress\_allow = 3.0e6;    n = 500;    % Deterministic variables  poisson = 0.15\*ones(n, 1);  length = 100\*ones(n, 1);  %% Random Variables  x1 = mvnrnd(x0(1), stddev(1), n);  x2 = mvnrnd(x0(2), stddev(2), n);  x3 = mvnrnd(x0(3), stddev(3), n);  x4 = mvnrnd(x0(4), stddev(4), n);  young = mvnrnd(x0(5), stddev(5), n);  density = mvnrnd(x0(6), stddev(6), n);  pressure = mvnrnd(x0(7), stddev(7), n);    %% Perform MCS  MCS\_max\_von\_mises = zeros(1, n);  ns = 0;    for i=1:n  X = [x1(i) x2(i) x3(i) x4(i) young(i) density(i) pressure(i) poisson(i) length(i)];  tic  MCS\_max\_von\_mises(i) = Arch\_bridge\_2D(X);  toc  fprintf('Finish step %d \n', i);  if MCS\_max\_von\_mises(i) <= stress\_allow  ns = ns + 1;  end    end  %% Calculate Reliability  R\_stress\_MCS = ns/n;    figure()  histogram(MCS\_max\_von\_mises, 'Normalization', 'pdf')  xlabel('Max stress [Pa]','fontsize',25)  ylabel('PDF','fontsize',25)  title('PDF of Stress','fontsize',25)  saveas(gcf, 'Hist of stress\_2D.png')  saveas(gcf, 'Hist of stress\_2D.fig')  save('2D\_MCS\_results\_normal.mat', 'MCS\_max\_von\_mises', 'R\_disp\_MCS', 'R\_stress\_MCS'); |
| %% Expansion Method (SORM)  clc; clear all; close all  disp('================ Start SORM ================')  nr = 7; nc = 1; % # of constraints  %% Values  % x0 = [7, 15, 2.5, 30, 20, 2000, 500]; % 2D initial design  x0 = [7 13.340319 2.002096 24 20 2000 500]; % 2D DO  stddev = [0.3, 0.6, 0.1, 0.8, 1.0, 100, 25];    stress\_allow = 3.0e6;  length = 100;  poisson = 0.15;    mu = x0;  sigma = stddev;  %% Perform Expansion Method (SORM)  max\_von\_mises = Arch\_bridge\_2D([mu poisson length]);  G = max\_von\_mises;  dGdx = zeros(nc, nr);    for kc=1:nc  fprintf('Start G(%d)!\n', kc);  for j=1:nr  delta\_x = zeros(1, nr);  delta\_x(j) = 1e-4 \* mu(j);  x\_plus\_delta\_x = mu + delta\_x;  max\_von\_mises\_plus\_delta\_x = Arch\_bridge\_2D([x\_plus\_delta\_x poisson length]);  G\_plus\_delta\_x = max\_von\_mises\_plus\_delta\_x;  dGdx(kc, j) = (G\_plus\_delta\_x(kc) - G(kc))/delta\_x(j);  fprintf('Finish variable %d\n', j);  end  for ii=1:nr  for jj=1:nr  ddGddx(ii, jj) = (dGdx(kc, ii) - dGdx(kc, jj))/(1e-3);  end  end  end    %% Approximate mean & standard deviation  mu\_G\_SORM = G + 1/2\*diag(ddGddx)'\*(sigma'.^2);  sigma\_G\_SORM = sqrt(dGdx\*diag(sigma.^2)\*dGdx' + 1/2\*sigma.^2\*ddGddx\*sigma'.^2);    %% Reliability (SORM)  R\_stress\_SORM = normcdf(stress\_allow, mu\_G\_SORM, sigma\_G\_SORM);    n\_sample = 1000;  % About Max von mises  x2 = linspace(mu\_G\_SORM - 4\*sigma\_G\_SORM, mu\_G\_SORM + 4\*sigma\_G\_SORM, n\_sample);  y2\_p = normpdf(x2, mu\_G\_SORM, sigma\_G\_SORM);  y2\_c = normcdf(x2, mu\_G\_SORM, sigma\_G\_SORM);    %================================ FORM ===================================  %% Expansion Method (FORM)  disp('================ Start FORM ================')  %% Perform Expansion Method (FORM)  max\_von\_mises = Arch\_bridge\_2D([mu poisson length]);  G = max\_von\_mises;  dGdx = zeros(nc, nr);    for i=1:nc  fprintf('Start G(%d)\n', i);  for j=1:nr  delta\_x = zeros(1, nr);  delta\_x(j) = 1e-4 \* mu(j);  x\_plus\_delta\_x = mu + delta\_x;  max\_von\_mises\_plus\_delta\_x = Arch\_bridge\_2D([x\_plus\_delta\_x poisson length]);  G\_plus\_delta\_x = max\_von\_mises\_plus\_delta\_x;  dGdx(i, j) = (G\_plus\_delta\_x(i) - G(i))/(delta\_x(j));  fprintf('Finish variable %d\n', j);  end  end      %% Approximate mean & standard deviation  mu\_G\_FORM = G';  sigma\_G\_FORM = sqrt(dGdx\*diag(sigma.^2)\*dGdx');    %% Reliability (FORM)  R\_stress\_FORM = normcdf(stress\_allow, mu\_G\_FORM, sigma\_G\_FORM);    % About Max von mises  x4 = linspace(mu\_G\_FORM - 4\*sigma\_G\_FORM, mu\_G\_FORM + 4\*sigma\_G\_FORM, n\_sample);  y4\_p = normpdf(x4, mu\_G\_FORM, sigma\_G\_FORM);  y4\_c = normcdf(x4, mu\_G\_FORM, sigma\_G\_FORM);    %% Load MCS results  clear max\_disp max\_von\_mises  disp('================ Load MCS results ================')  load('2D\_MCS\_results\_\_DO.mat')    %% Plot figures (MCS FORM SORM)  figure()  histogram(MCS\_max\_von\_mises,'Normalization','pdf')  hold on;  plot(x4, y4\_p, '-r', x2, y2\_p, '-b')  xlabel('Max stress [Pa]','fontsize',25)  ylabel('PDF','fontsize',25)  title('PDF of Stress','fontsize',25)    legend('MCS', 'FORM', 'SORM')  % legend('FORM', 'SORM')  saveas(gcf,'PDF of stress\_coupled.png')  saveas(gcf,'PDF of stress\_coupled.fig')    figure()  cdfplot(MCS\_max\_von\_mises);  hold on;  plot(x4, y4\_c, '-r', x2, y2\_c, '-b')  xlabel('Max stress [Pa]','fontsize',25)  ylabel('CDF','fontsize',25)  title('CDF of Stress','fontsize',25)    legend('MCS', 'FORM', 'SORM')  % legend('FORM', 'SORM')  saveas(gcf,'CDF of stress\_coupled.png')  saveas(gcf,'CDF of stress\_coupled.fig') |