The Hoover dam

Linear elasticity: Axisymmetric structures

Chalmers University of technology

LACERDA ANTONIO – BROUSSEAU PAUL

Lacerda Antonio
Brousseau Paul

Problem description

The Hoover dam was build in 1931-1936 and is located in the United States of America. He backs up the water of the Colorado river and forms the lake Mead. It is a curved gravitational dam.

Figure 1 - Photographe of the Hoover dam

We will assume that the only loads on the dam are the water pressure and its own weight (gravity). The air pressure equals zero. Furthermore, the reaction of the ground is unknown.
This problem will be resolved as a 2D axisymmetric problem. This dam is curved, that's why we will only focus on one cross section of the hoover dam because we assume that each cross-section behaves identically.

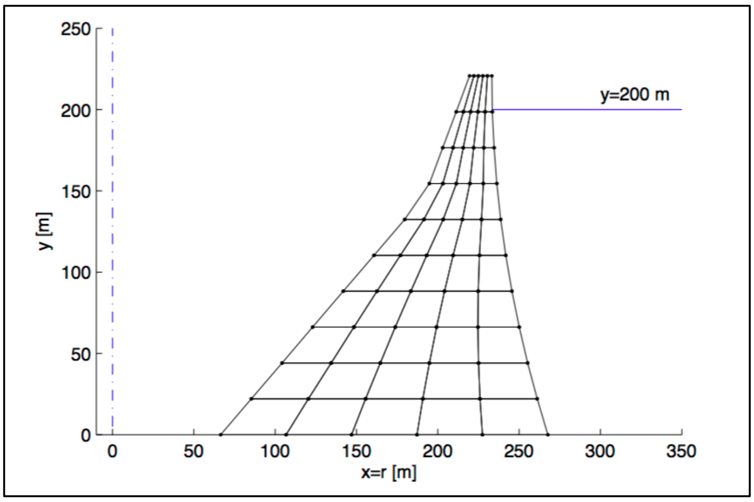

Figure 2 - One cross-section of the Hoover dam

In addition, the displacement for y=0 equals zero. All the others are unknown and are sought for. Seeing displacement when the dam is loaded permits to have an idea if our solution is right or wrong.
Finally, we will determine the full stress tensor in all the area of the cross-section of the dam and for each element the solution will be plot to see the distribution of the stress in all the area.

Method

In order to compute the solution, we need to start with the strong form which represents the equilibrium inside the area of the cross-section of the dam to arrive to the weak form. Moreover, we know that the problem is two dimensional and axisymmetric.

The weak form

We consider the domain in the r-z plane (figure 2). The equilibrium inside the domain is:

$$-(\widetilde{\nabla}_1 - \widetilde{\nabla}_0)^T (rD(\widetilde{\nabla}_1 + \widetilde{\nabla}_0)u) = rb$$

$$\text{with } \widetilde{\nabla}_1 = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ 0 & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \end{bmatrix}$$

We multiply this equation by a test function v: $v = [v_r \;\; v_z]^T$ and integrate over A :

$$-v^T(\widetilde{\nabla}_1 - \widetilde{\nabla}_0)^T (rD(\widetilde{\nabla}_1 + \widetilde{\nabla}_0)u) = v^T rb$$

$$-\int_A v^T \widetilde{\nabla}_1{}^T r\sigma dA + \int_A v^T \widetilde{\nabla}_0{}^T r\sigma dA = \int_A v^T rbdA \qquad (1)$$

We can notice that $\sigma = D\varepsilon = D\widetilde{\nabla}u$ with D a symmetric matrix.
Or, the first term of (1) equals to:

$$-\int_A v^T \widetilde{\nabla}_1{}^T r\sigma dA = -\int_A [v_r \;\; v_z] \begin{bmatrix} \frac{\partial}{\partial r} & 0 & 0 & \frac{\partial}{\partial z} \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \end{bmatrix} r \begin{bmatrix} \sigma_{rr} \\ \sigma_{\varphi\varphi} \\ \sigma_{zz} \\ \sigma_{rz} \end{bmatrix}$$

$$= \int_A r(v_r \frac{\partial \sigma_{rr}}{\partial r} + v_r \frac{\partial \sigma_{rz}}{\partial z} + v_z \frac{\partial \sigma_{zz}}{\partial z} + v_z \frac{\partial \sigma_{rz}}{\partial r})dA - \int_A v_r(\sigma_{rr} + \sigma_{rr})dA$$

We now weed to reduce the regularity requirements by integration by part or by using the Gauss theorem with cylindrical coordinates.
If g is a vector and ϕ a scalar :

$$if\ g = [\varphi \;\; 0] \rightarrow \int_A \phi \frac{\partial \varphi}{\partial r} dA = \int_L \phi\, \varphi n_r dL - \int_A (\frac{\partial \phi}{\partial r} + \frac{\varphi}{r})dA$$

$$if\ g = [0 \;\; \varphi] \rightarrow \int_A \phi \frac{\partial \varphi}{\partial r} dA = \int_L \phi\, \varphi n_z dL - \int_A (\frac{\partial \phi}{\partial z})dA$$

Therefore,

$$\int_A v^T \tilde{\nabla}_1{}^T r\sigma dA = r\left(\oint_L (v_r\sigma_{rr}n_r + v_r\sigma_{rz}n_z + v_z\sigma_{zz}n_z + v_z\sigma_{rz}n_r)dL - \int_A (\frac{\partial v_r}{\partial r}\sigma_{rr} + \frac{\partial v_z}{\partial z}\sigma_{zz} + \frac{\partial v_z}{\partial z}\sigma_{rz} + \frac{\partial v_z}{\partial r}\sigma_{rz})dA - \int_A \frac{v_r(\sigma_{rr} + \sigma_{rz)}}{r}dA\right) + \int_A v_r(\sigma_{rr} + \sigma_{rz)}dA$$

Thus, we can rearrange this equation:

$$-\int_A v^T\tilde{\nabla}_1{}^T r\sigma dA = -\oint_L v^T r\begin{bmatrix}\sigma_{rr} & \sigma_{rz}\\ \sigma_{rz} & \sigma_{zz}\end{bmatrix}\begin{bmatrix}n_r\\ n_z\end{bmatrix}dL + \int_A (\tilde{\nabla}_1 v)^T r\sigma dA \qquad (2)$$

Because,

$$(\tilde{\nabla}_1 v)^T r\sigma = \left(\begin{bmatrix}\frac{\partial}{\partial r} & 0\\ 0 & 0\\ 0 & \frac{\partial}{\partial z}\\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r}\end{bmatrix}\begin{pmatrix}v_r\\ v_z\end{pmatrix}\right) r\begin{bmatrix}\sigma_{rr}\\ \sigma_{\varphi\varphi}\\ \sigma_{zz}\\ \sigma_{rz}\end{bmatrix} = \begin{bmatrix}\frac{\partial v_r}{\partial r}\\ 0\\ \frac{\partial v_z}{\partial z}\\ \frac{\partial v_r}{\partial r} + \frac{\partial v_z}{\partial z}\end{bmatrix} r\begin{bmatrix}\sigma_{rr}\\ \sigma_{\varphi\varphi}\\ \sigma_{zz}\\ \sigma_{rz}\end{bmatrix}$$

$$= r(\frac{\partial v_r}{\partial r}\sigma_{rr} + \frac{\partial v_z}{\partial z}\sigma_{zz} + (\frac{\partial v_r}{\partial r} + \frac{\partial v_z}{\partial z})\sigma_{rz}$$

Therefore, with (1) and (2), we have:

$$\int_A v^T \tilde{\nabla}_0{}^T r\sigma dA + \int_A (\tilde{\nabla}_1 v)^T r\sigma dA + \oint_L v^T r\begin{bmatrix}\sigma_{rr} & \sigma_{rz}\\ \sigma_{rz} & \sigma_{zz}\end{bmatrix}\begin{bmatrix}n_r\\ n_z\end{bmatrix}dL = \int_A v^T rbdA$$

Or we know that $\tilde{\nabla} = \tilde{\nabla}_1 + \tilde{\nabla}_0$ and $(\tilde{\nabla}_1 v)^T = \tilde{\nabla}_1{}^T v^T$ and $\begin{bmatrix}\sigma_{rr} & \sigma_{rz}\\ \sigma_{rz} & \sigma_{zz}\end{bmatrix}\begin{bmatrix}n_r\\ n_z\end{bmatrix} = S.n = t$ and $\sigma = D\tilde{\nabla}N$. Therefore:

$$\int_A (\tilde{\nabla} v)^T r\sigma dA = \int_A v^T rbdA + \int_L v^T rtdL$$

<u>Note:</u> t is not totally known. All the other variables are known. Furthermore, in order to have the entire weak form, we need to introduce natural boundary conditions.

Then, the general case of the weak form is:

$$\begin{cases}\int_A (\tilde{\nabla} v)^T r\sigma dA = \int_A v^T rbdA + \int_{Lh} v^T rhdL + \int_{Lg} v^T rtdL \quad (3)\\ u = g \text{ on } Lg\end{cases}$$

The boundary conditions for the dam simulation will be explained in detail later (see Boundary conditions).

Finite element formulation

In order to compute our problem, we need to introduce the finite element formulation. That's why we introduce the approximation $u(r,z) = N(r,z)a$ with $a = \begin{bmatrix} u_{r1} \\ u_{z1} \\ \vdots \\ u_{rn} \\ u_{zn} \end{bmatrix}$.

Furthermore, the Galerkin restriction of the weight function is : $v(r,z) = N(r,z)c$ where c is an arbitrary coefficient. This restriction combined with the equation (3) leads to the discrete FE-equations:

$$Ka = f_b + f_c$$

With $\begin{cases} K = \int_A B^T rDBdA \\ f_b = \int_{L_h} N^T rhdL + \int_{L_g} N^T rtdL \\ f_l = \int_A N^T rbdL \end{cases}$

We need to deduce B. We have:

$$Ka = f_b + f_l = \int_A B^T rDBadA = \int_{Lh} N^T rhdL + \int_{Lg} N^T rtdL + \int_A N^T rbdA$$

By taking c= $[c_F \quad c_C]$ where c_F *and* c_C are equal to 1, we can say that: $f_b + f_l = \int_A (\tilde{\nabla} N)^T rD\tilde{\nabla} vdA$

Moreover, $u(r,z) = N(r,z)a$, so:

$$\int_A B^T rDBadA = \int_A (\tilde{\nabla} N)^T rD\tilde{\nabla} NadA$$

Thus, we can deduce that $B = (\tilde{\nabla} N)$ and $B^T = (\tilde{\nabla} N)^T$. We can notice that B is a 4x(2n) matrix:

$$B = \tilde{\nabla} N = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ \frac{1}{r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \end{bmatrix} \begin{bmatrix} N_1(r,z) & 0 & N_2(r,z) & 0 & \cdots & N_n(r,z) & 0 \\ 0 & N_1(r,z) & 0 & N_2(r,z) & \cdots & 0 & N_n(r,z) \end{bmatrix}$$

So,

$$B = \begin{bmatrix} \frac{\partial N_1(r,z)}{\partial r} & 0 & \cdots & \frac{\partial N_n(r,z)}{\partial r} & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \frac{N_1(r,z)}{r} & \frac{\partial N_1(r,z)}{\partial z} & \cdots & \frac{N_n(r,z)}{r} & \frac{\partial N_n(r,z)}{\partial z} \\ \frac{\partial N_1(r,z)}{\partial z} & \frac{\partial N_1(r,z)}{\partial r} & \cdots & \frac{\partial N_n(r,z)}{\partial z} & \frac{\partial N_n(r,z)}{\partial z} \end{bmatrix}$$

Boundary conditions

We have now to determine the boundary conditions for the problem. We can notice that we have different form of boundary conditions. First, there are the loads on the dam from the water pressure and the gravity of the dam. We choose the air pressure (atmospheric pressure) equals zero. To do this, we divide the dam boundary in three parts.

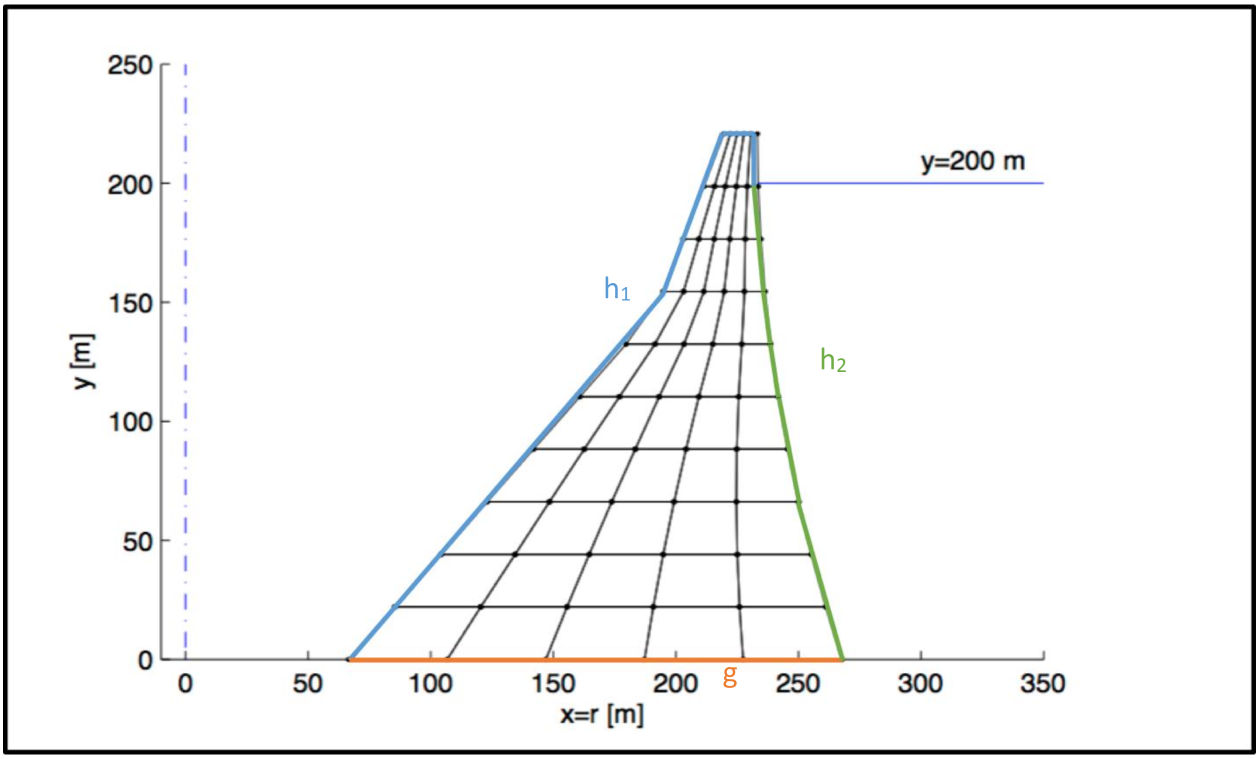

Figure 3 - Definition of the boundaries

Then the boundaries conditions are:

- $b = -\rho_{concrete} g$ on A.
- $h = t_{h_2} = \rho_{water} g(h - y)$ on L_{h2} for $y \leq 200$
- $h = h_1 = 0$ on L_{h1}
- $u = 0$ on L_g

Nodal displacement arrangement

We have already seen that: $Ka = f_b + f_c \quad (4)$

with $\begin{cases} K = \int_A B^T rDBdA \\ f_b = \int_{L_h} N^T rhdL + \int_{L_g} N^T rtdL \\ f_l = \int_A N^T rbdL \end{cases}$

The boundary conditions combine to the equation (4) lead to:

$$Ka = K\begin{bmatrix} a_F \\ a_c \end{bmatrix} = \int_{L_{h1}} N^T rh_1 dL + \int_{L_{h2}} N^T rt_{h_2} dL + \int_{L_g} N^T rtdL + \int_A N^T rbdL + \int_{L_g \cap L_{h2}} N^T rt' dL + \int_{L_{h1} \cap L_{h2}} N^T rh' dL = \begin{bmatrix} f_F \\ f_c \end{bmatrix}$$

In this step, we will introduce another assumption. This is related to the number of elements in the simulation. When the number of elements is big enough, we can consider $L_g \cap L_{h2}$ and $L_{h1} \cap L_{h2}$ to be an empty set, which in allow us to neglect the effect of those points on the boundary conditions. This assumption will simplify the expression above and we will obtain:

$$Ka = K\begin{bmatrix} a_F \\ a_c \end{bmatrix} = \int_{L_{h1}} N^T rh_1 dL + \int_{L_{h2}} N^T rt_{h_2} dL + \int_{L_g} N^T rtdL + \int_A N^T rbdL = \begin{bmatrix} f_F \\ f_c \end{bmatrix} \quad (5)$$

Because a can be separate: $a = \begin{bmatrix} a_F \\ a_c \end{bmatrix}$ with L_{h1}, L_{h2} belonging to a_F, the free displacement and L_g belonging to a_c the constrained displacement. Thus, f_F is the load on a_F and f_c is the load on a_c. We can also separate N : $N = \begin{bmatrix} N_F \\ N_c \end{bmatrix}$ and K: $K = \begin{bmatrix} K_{FF} & K_{FC} \\ K_{CF} & K_{CC} \end{bmatrix}$. Then, we can say that:

$$\begin{cases} f_F = \int_{L_{h1}} {N_F}^T rh_1 dL + \int_{L_{h2}} {N_F}^T rt_{h_2} dL + \int_{L_g} {N_F}^T rtdL + \int_A {N_F}^T rbdL \\ f_c = \int_{L_{h1}} {N_C}^T rh_1 dL + \int_{L_{h2}} {N_C}^T rt_{h_2} dL + \int_{L_g} {N_C}^T rtdL + \int_A {N_C}^T rbdL \end{cases}$$

On L_g, N_F equals zero because the nodes on L_g are constrained and not free. Therefore, f_F is totally known. We can also define f'_C the known part of f_C and r, the unknown. Therefore:

$$\begin{cases} f_F = \int\limits_{L_{h1}} {N_F}^T r h_1 dL + \int\limits_{L_{h2}} {N_F}^T r t_{h_2} dL + + \int\limits_{A} {N_F}^T r b dL \\ f_c = f'_C + r \end{cases}$$

With $\begin{cases} f'_C = \int_{L_{h1}} {N_C}^T r h_1 dL + \int_{L_{h2}} {N_C}^T r t_{h_2} dL + \int_{A} {N_C}^T r b dL \\ r = \int_{L_g} {N_C}^T r t dL \end{cases}$

Finally, we can establish the equation from which the free variables can be solved (a) and the equation for the support reaction r (b), which would be calculated in a post-processing step:

$$\begin{cases} K_{ff} a_f = f_c - K_{fc} a_c & (a) \\ r = K_{cc} a_c + K_{cf} a_f - {f'}_C & (b) \end{cases}$$

Thus, we are now able to compute the solution, using Matlab and Calcem.

Results and discussion

We can now use Matlab, and more precisely Calcem to solve the FE-problem introduced above. The Matlab source code is uploaded along the report on PingPong. To analyse the results, we will plot the largest principal stress to investigate the risk for cracking in the dam and the largest compressive stress in the cross-section. Finally, displacement field of the area will be shown.

Stress distribution on the dam

Displacement

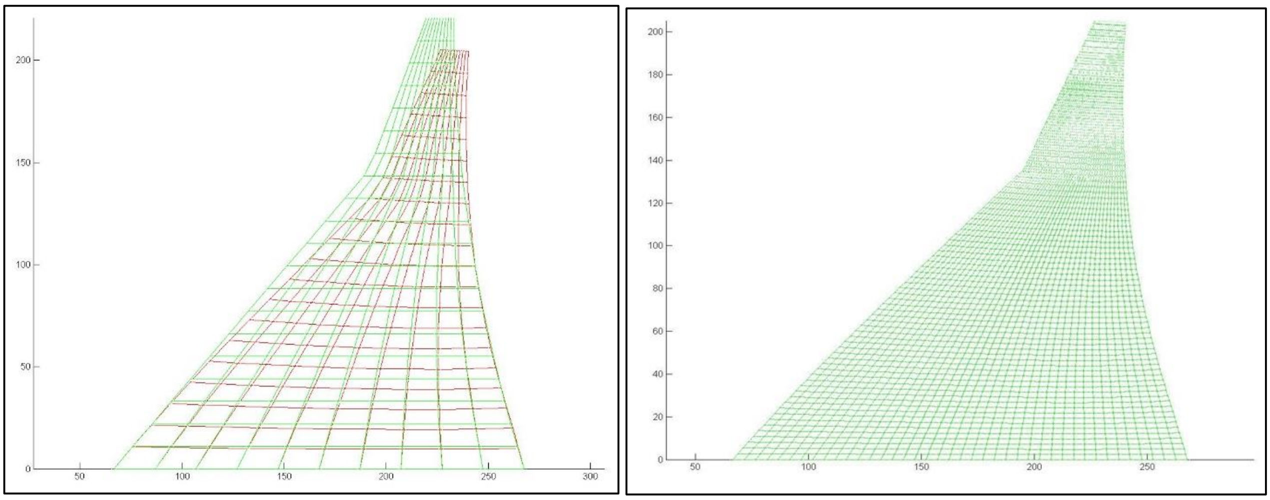

Figure 4 : Displacement field for the dam calculated using the help of the Calfem program in Matlab. The plot to the left is made under 20 x 10 elements (n° of elements along its height x n° of elements along its thickness). For the right side, we used 80x40. The units along the x(r) and y-axis is meters. In the left plot, the displacement field is in red while in green we have the original shape of the dam. In the picture to the right, we only have the displacement field in green.

The displacement is the best simulation, if compared to the compressive and principal stress, to see if our solution is in agreement with the reality or not. Explaining its importance and why it is the first plot to be discussed[1].

First, we can see that both our results agree with each other. Furthermore, the boundary condition on the bottom (Lg) is verified, i.e. there is no displacement. Moreover, we can see compression in the area near the middle and bottom right, were there would be water (bellow 200 for our case) which agrees with the theory and we see tension in the top, which, again, is expected. In addition, we can see that the dam is curving towards the right. This does agree with reality, since, as the water is not being applied in the entire right side, the top will move to the right, as the bottom and middle are being pressed upon by the water. Finally, there is no discontinuity is the displacement, which is a requirement for this simulation be accepted.

The maximum displacement, for both simulations are similar (there is a small difference), which is $0.00725\ m$. The minimum displacement, i.e. compression, is of $-0.01656\ m$.

[1] For example, when coding the MatLab program, we used the displacement simulation in other to find out where we were making our mistake, which could not be seen by the two stress plots. In our example, there was a problem in the boundary conditions, which led to a very unrealistic result of the displacement simulation.

Lacerda Antonio
Brousseau Paul

Although we do obtain good results in this simulation, before confirming that the Matlab code and that the assumptions made in the simulation are correct, we need to observe the compressive stress and the displacement simulations.

Largest principal stress

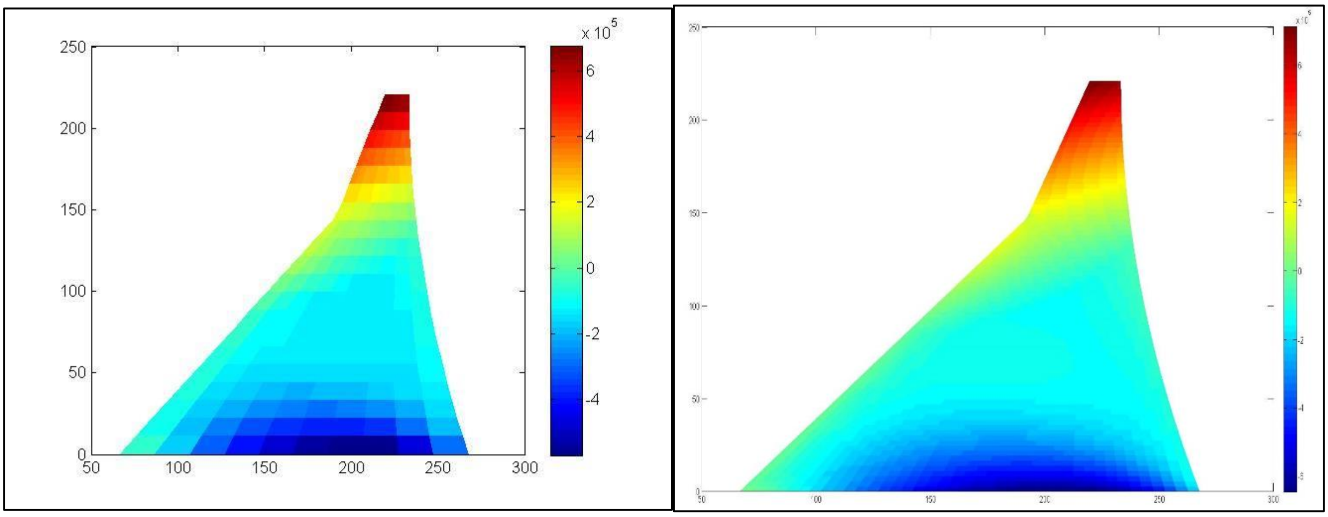

Figure 5 : The principal stress field in the Dam. As before, in the plot in the left side, we used 20x10 elements and in the one at the right side, we used 80x40 elements. The x(r)-axis and the y-axis are in meters and the stress scale bar is in Pascal. We can see that the maximum tensile stress is around 6.10^5 Pa.

We can notice that the graph with the smallest elements, i.e. with the biggest amount of elements, is more continuous. In addition, the values are similar in both simulations. Therefore, this result is independent of the size of the element. Moreover, we can say that this result seems to agree with the reality. This is because the largest principal stress is located at the top, where we will have the maximum displacement (tension) and is minimal at the bottom, where we have compression. In the center and especially at the left side of the dam, we can see that that the modulus of the principal stress is the smallest among the other elements of this simulation. This can be explained by the displacement of the points in left side of the dam, which is the smallest among the rest.

We can see here that under tensile stress the maximum stress would be of around 6.10^5 Pascal. The 28-day compressive strength for a structure lightweight concrete is around 2500 psi, i.e. $17{,}2.10^6$ Pascal (The Aberdeen Group, 1981). The density of such concrete can go up to the range of $1950\ Kg.m^{-3}$ (The Aberdeen Group, 1981). The concrete used here has a higher density than a lightweight concert, which would, in turn, increase its 28-day compressive strength. Moreover, the tensile ultimate strength of concrete is around 10% to 15% of its compressive strength (Assakkaf, 2002). Therefore, we can conclude that, with a maximum tensile stress of 6.10^5 Pa, the structure would not fail due to excess of tensile stress.

Moreover, finding a smaller result for the maximum tensile stress in the dam than for the compressive tensile stress (as we will see afterwards) is in good agreement with the reality. This is true due to, beyond the tensile strength of concrete being extremely small if compared to the compressive strength, the fact that it is usually considered zero for practical applications[2], i.e.

[2] Once a crack is created in concrete, the stress necessary for its propagation is a much smaller than the one for its initiation. Therefore, when it starts it will most likely, with time, propagate to the entire tensile area.

when building a dam. This implies that the shape of the dam will be designed with the intention of reducing the maximum tensile stress and the volume of the dam under tensile stress.

Compressive stress

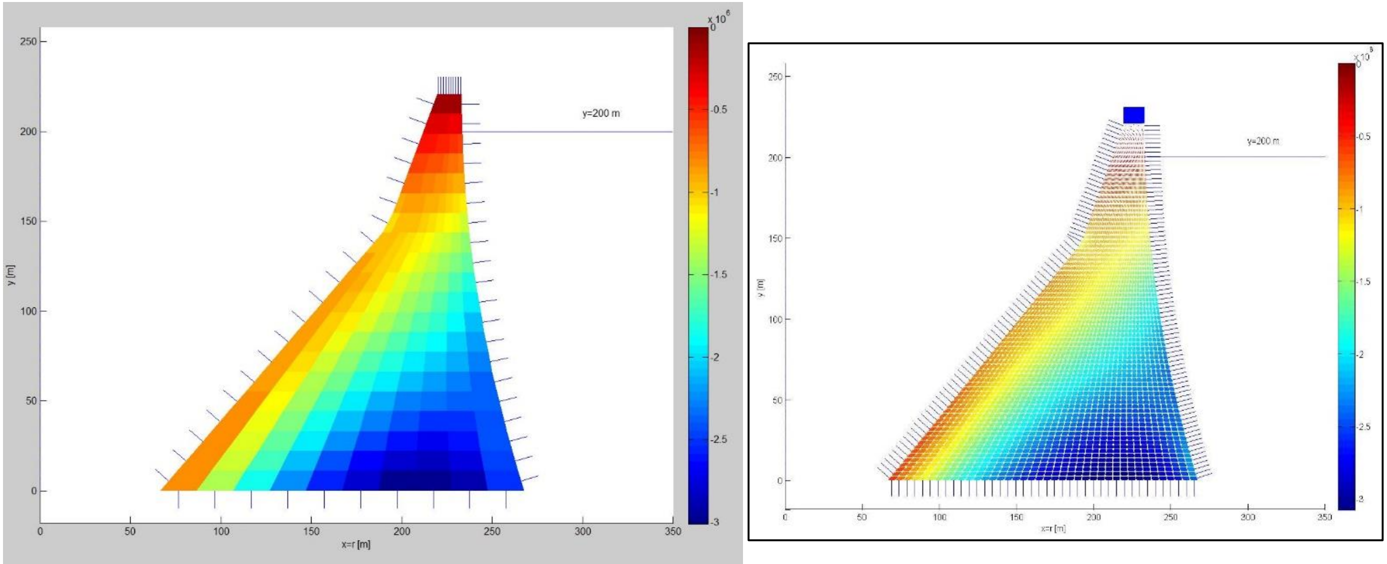

Figure 6 : Compressive stress field for the dam. Same number of elements as before and we used the same scale as for the principal stress field. The modulus of the compressive stress attains a maximum of 3.10^6 *Pa.*

As observed for the largest principal stress, we can see that the simulation using smaller elements is more continuous and both graphs are in agreement with each other. Therefore, this result is independent of the size of the element. We can see here that the compressive stress at the top is zero and its modulus is at its maximum at the bottom right of the dam. The latter is easily explained by the fact that the points at the bottom right of the dam will be the ones suffering more compression than the others. The former can be understood as if the top suffers no compression, which is true, since the displacement in those points will be tension. On the left side, especially in the bottom, we can see that some points seems to suffer almost no compression. This can be understood as if the size of the dam is big enough to make the compression due to the pressure of the water not to be seen in the left side. The assumption made for the air pressure equals zero might explain the low compression on this side.

Moreover, the assumption u=0 for y=0 is far away from the reality. Indeed, the dam is tamping in the ground, so the assumptions and boundary conditions at the bottom should not be zero for the displacement. This is because of the porosity of the ground, which will in turn allow some water to penetrate it and affect the load at the bottom of the dam. In addition, due to the unknown behavior at the edges of the dam, i.e. in the section where the dam meets the “mountain”, we are not able to to forecast the stress on the two sides of the dam (we assume that every cross-section is the same).

Finally, the maximum of the modulus of the compressive stress is 3.10^6. This is well within of the limits for the concrete that was used in the dam (assuming that is has a higher compressive strength then a structural lightweight concrete).

Conclusion

By observing our three simulations, we can conclude that the Mat lab program works and it is in agreement with what one would expect to happen in a real case scenario for the stress and displacement field in a cross section of the Hoover Dam.

Moreover, we can state that by increasing the number of elements in used by the FEM simulation one would obtain a more continuous plot and a reduced effect of the points in the boundary for which two different conditions apply, e.g. where part of the element is under pressure of water while the other part is under pressure of the air (neglected in this report).

Appendix

```
close all
clear all


% Initialization of constants

rhowater = 1000;  %%
g = 9.81;  %% m/s^2
rhoconcrete = 2300;  %%
poisson = 0.18;
E = 22e9;  %%
nheight = 20  ;  %% m
nthick = 10;   %% m
waterheight = 200;  %% m
Patm = 1e5 ; %%


% type  3 - axysimmetric

D = hooke(3,E,poisson);

% recovering positions and normal values

[Ex,Ey,Edof,Boundarygeom,Boundarydof]=hoover2d(nthick,nheight);

% number of boundary segments

nbound = size (Boundarydof);

% number of degrees of freedom

ndofs = max(max(Edof(:, 2:end)));


f_b = zeros(ndofs,1);

% matrix N that will be used later to calculate f_b for the boundary where
% we have water preassure

N = [ 1,0;
    0,1;
    1,0;
    0,1;
    ];

% calculating the stresses/ boundary conditions for water side. As for the
% air side and goround we consider the preassure to be vector T to be zero
% we don't need to calculate them, as it was already done when we
% initialized f_b

for n = 1:nbound(1)

    x1 = Boundarygeom(n,1);
    x2 = Boundarygeom(n,3);
    y1 = Boundarygeom(n,2);
```

```
    y2 = Boundarygeom(n,4);
    nx = Boundarygeom(n,5);
    ny = Boundarygeom(n,6);

    lf =sqrt((x1-x2)^2+(y1-y2)^2);

    Normal = [nx;ny];

    dofs = Boundarydof(n,:);
    fbf = zeros(4,1);


    % for water side

    if (ny ~= 1) &&(ny ~= -1) && (y1 || y2 < waterheight) && (sign(nx)==1)

        % 1 by 2 matrix with the traction on the water side, it will be
        % zero otherwise

        T  = -((rhowater*g*(waterheight-(y1+y2)/2)))*(Normal);
        fbf = (lf*(N)*(x1+x2)*T)/4;

    end

    f_b(dofs) = f_b(dofs) + fbf(:);

end

% boundary conditions for the botton

bc = zeros (nthick*2+2,2);
bc (:,1) = 1:(nthick*2 +2);

% number of elements

nelement = size (Ex);

K = zeros (ndofs,ndofs);
F = zeros (ndofs,1);

% Calculating the K and F matrix for each element

for k = 1: nelement(1)


    ex = Ex(k,:);
    ey = Ey(k,:);
    eq = [0;0];

    % the bodyforce

    eq(2) = -rhoconcrete*g;

    % n x n integration points where n the number of gauss poits
    % xa and ya = 0 ,nxa = 0 and nya = 1

    ep = [0;0;0;1;2];

    [ke,fe]=axi4e(ex,ey,ep,D,eq);
```

```
    [K,F] = assem(Edof(k,:),K,ke,F,fe);

end

% Calculating the displacements for each element

F = F + f_b;

[Af] = solveq(K,F,bc);

% This will be usde to plot the unmodified dam
sizeF = size(F);
Fzero = zeros (sizeF(1),sizeF(2));

[Afzero] = solveq(K,Fzero,bc);

Edzero = extract (Edof,Afzero);

% Extract the displacement for each element

Ed = extract (Edof,Af);

Sprincipal = zeros(3,nelement(1));

% Calculating the principal stresses in each element

for l = 1: nelement(1)


    % xa and ya = 0

    ep = [0;0;0;1;2];


    % here we obtain a matrix with the stress associated with each point

    [es,~,~]=axi4s(Ex(l,:),Ey(l,:),ep,D,Ed(l,:));

    % here we calculate the general matrix by taking the mean of all gauss
    %
    sprincipal = [mean(es(:,1)),0,mean(es(:,4));
        0,mean(es(:,3)),0;
        mean(es(:,4)),0,mean(es(:,2));
        ];


    Sprincipal(:,l) = eig(sprincipal);

end

% maximum stress    ;

max_Sprincipal  = zeros (nelement(1),4);

max_Sprincipal(:,1) = (Sprincipal(3,:));

% minimum stress
```

```
min_Sprincipal  = zeros (nelement(1),4);

min_Sprincipal(:,1) = (Sprincipal(1,:));


figure (1)


fill (Ex',Ey',min_Sprincipal');
shading flat
colorbar
colormap(jet);
figure (2)

fill (Ex',Ey',max_Sprincipal');
shading flat
colorbar
colormap (jet)

figure(3)

hold on

plotpar = [1 4 1];
eldisp2(Ex,Ey,Ed,plotpar,1000)

plotpar = [1 2 1];
eldisp2(Ex,Ey,Edzero,plotpar,1000)

hold off
```

[*Published with MATLAB® R2014b*]()

Bibliography

Assakkaf, H. (2002). Part 1 - Concrete Design and Analysis. *Materials and Mechanics of Bending*. Maryland. Fonte: www.assakkaf.com/courses/ence355/lectures/part1/chapter1b.pdf

The Aberdeen Group. (1981). *At the high end of the lightweight concrete spectrum : Structural lightweight concrete*. Fonte: www.concreteconstruction.net/Images/Structural%20Lightweight%20Concrete_tcm45-345994.pdf