"Implementation of the Pushover Analysis in MatLab for 2D frames"

Luis Fernando Verduzco Martínez [1], Jaime Moises Horta Rangel [2], Miguel Ángel Pérez Lara y Hernández [3], Juan Bosco Zaragoza [4]

[1] luiz.verduz@gmail.com, [2] horta@uaq.mx, [3] migperez@uaq.mx, [4] bosco@uaq.mx

Faculty of Engineering, Autonomous University of Quretaro, Santiago de Querétaro, Mexico

Overview

This software is the implementation of the Pushover analysis for structural plane frames, and it has been fully developed in MatLab as a function. Such function is based entirely on flexure hinges formations at the ends of the structural elements composing a structural plane frame. A uniform seismic load incrementation is carried out on each step until a collapse or stiffness degradation criteria is reached. Such function has proved to have a great potential for its implementation in teaching the Pushover method in structural mechanics and applied sciences through the simulation of collapse mechanisms. Not only $P-\Delta$ collapse graphics are obtained, but also the evolution of the collapse mechanism deformation of the structure and the seismic Collapse Safety Factors (CSF), including as well the evaluation of some basic Damage Indices (DI) such as the Inter-story Drift DI, the Inter-story Plastic Drift DI and the Deformation Based DI.

Keywords: Pushover, Plane Frames, Teaching, Pushover, Collapse Mechanisms , Damage Indices

1 The Pushover analysis

The *Pushover* analysis is used in structural engineering for many tasks when is required to evaluate the response of a structure in the plastic range behaviour, for instance, when evaluating performance of structures and estimating potential Damage States under certain load conditions. It consists of analysing a structure by incremental lateral loads imposed, defining step by step plastic-formations on the element's cross-section. The analysis is terminated when the collapse mechanism is reached, either by stating a certain state of damage for the structure or a degree of stiffness degradation. This way, seismic Collapse Safety Factors can be found **Fig. 1**. A condition of stiffness degradation can be imposed as (1) to stop the analysis process, where K_j is the last stiffness matrix, K_0 the initial under elastic behaviour and deg is a stiffness degradation factor (for which values between $0.003-0.005$ are recommended).

$$\frac{det(K_j)}{det(K_0)} < deg \tag{1}$$

During the analysis process, when a plastic formation is detected over one or various elements' cross-sections, then such elements are transformed to an equivalent structure in which such plastic formations are considered as articulations. Such articulations are simulated through the liberation of the DOF in such plastic formation location and its corresponding proper distribution of the loads and bending moments to the element's ends according to the new generated equivalent structural system due to such DOF liberation **Fig. 2**. This way, when both ends of an element have suffered plastic formations its equivalent structural system will correspond to the one shown in **Fig. 3**. The external equivalent plastic bending

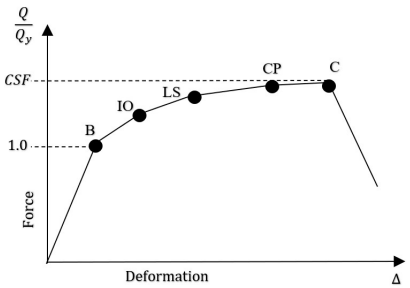

Figure 1. Force-Deformation digram representing the damage states (IO, LS, CP) through the collapse mechanism (C) of ductile structural frame. IO stands for Immediate Occupancy, LS for Life Safety, CP for Collapse Prevention according to [FEMA 273, 1997].

moments at each plastic formation location will remain constant through out the subsequent analysis steps such that the structure's stiffness and force-displacement capacity will be eventually degraded.

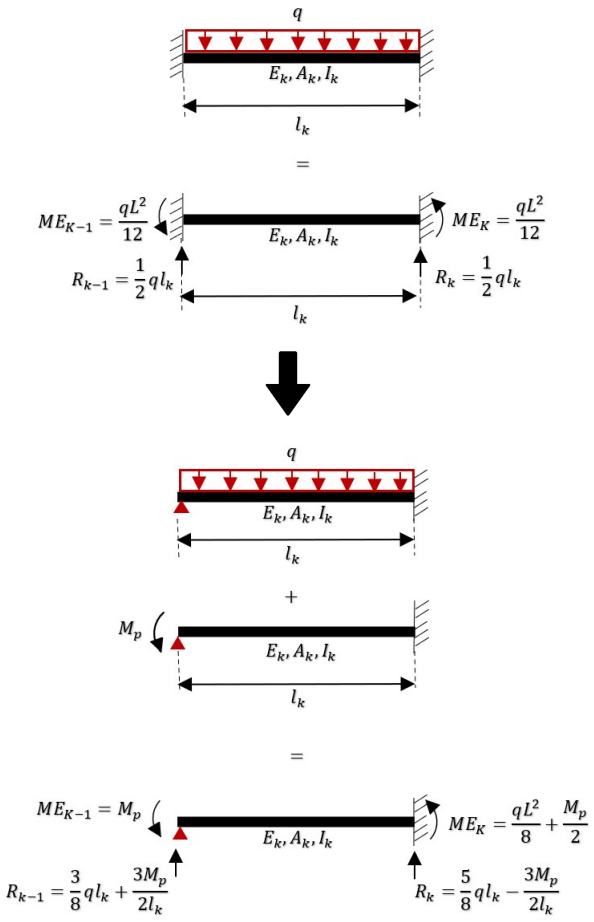

Figure 2. Equivalent structural system transformation when plastic formations at the ends of an element take place.

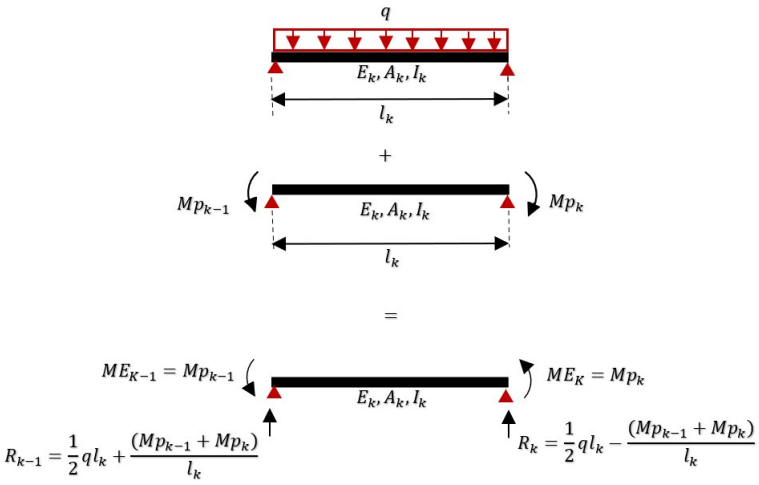

Figure 3. Equivalent structural system transformation when plastic formations at each of an element's ends takes place.

The stiffness matrix of a 2D structural element that has not yet suffered any plastic formation at any of its ends is represented as (2). On the other hand, the stiffness matrix for an equivalent structural system of an element with a plastic formation at its left end is (3):

$$[K] = \begin{bmatrix} \frac{EA}{l} & 0 & 0 & -\frac{EA}{l} & 0 & 0 \\ 0 & \frac{12EI_z}{l^3} & \frac{6EI_z}{l^2} & 0 & -\frac{12EI_z}{l^3} & \frac{6EI_z}{l^2} \\ 0 & \frac{6EI_z}{l^2} & \frac{4EI_z}{l} & 0 & -\frac{6EI_z}{l^2} & \frac{2EI_z}{l} \\ -\frac{EA}{l} & 0 & 0 & \frac{EA}{l} & 0 & 0 \\ 0 & -\frac{12EI_z}{l^3} & -\frac{6EI_z}{l^2} & 0 & \frac{12EI_z}{l^3} & -\frac{6EI_z}{l^2} \\ 0 & \frac{6EI_z}{l^2} & \frac{2EI_z}{l} & 0 & -\frac{6EI_z}{l^2} & \frac{4EI_z}{l} \end{bmatrix} \tag{2}$$

$$[K] = \begin{bmatrix} \frac{EA}{l} & 0 & 0 & -\frac{EA}{l} & 0 & 0 \\ 0 & \frac{3EI_z}{l^3} & 0 & 0 & -\frac{3EI_z}{l^3} & \frac{3EI_z}{l^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{l} & 0 & 0 & \frac{EA}{l} & 0 & 0 \\ 0 & -\frac{3EI_z}{l^3} & 0 & 0 & \frac{3EI_z}{l^3} & -\frac{3EI_z}{l^2} \\ 0 & \frac{3EI_z}{l^2} & 0 & 0 & -\frac{3EI_z}{l^2} & \frac{3EI_z}{l} \end{bmatrix} \tag{3}$$

Similarly, when the structural element system has suffered a plastic formation at both of its ends,

then its stiffness matrix takes the form of spar element in which only axial stiffness is considered (4):

$$[K] = \begin{bmatrix} \frac{EA}{l} & 0 & 0 & -\frac{EA}{l} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{l} & 0 & 0 & \frac{EA}{l} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

1.1 Damage Assessment

In order to evaluate peak dynamic deformation demand of structures at various performance levels, non-linear static analysis are required, so that a good estimation of damage may be determined for the structure life-cycle. The assessment of damage performance in structures is of great importance in the design stage of a structure, since it can be related with potential economical losses (cost of repairing, rehabilitation, etc.).

One of the most effective tools to estimate damage performance is through *Damage Indices (DI)*. A vast number of Damage Indices have been proposed and developed based on structural properties (response) or dynamical properties of the structure. Such indices may predict the level of degradation state of a structure and therefore its vulnerability. For this function, the following three non-cumulative DI were considered: *the Inter-story Drift DI* (5), *the Plastic Inter-story Drift DI* (6) and *the Deformation Based DI* (7). Such DI can reflect quite well the state of a structure at his last stage of collapse and are among the most detailed DI used and of quite simplicity for application [Makhloof et al., 2021].

$$DI_{drift} = \frac{\Delta_{max}}{H} \tag{5}$$

$$DI_{p-drift} = \frac{\Delta_{max} - \Delta_y}{H} \tag{6}$$

$$DI_{\mu} = \frac{\Delta_{max} - \Delta_y}{\Delta_u - \Delta_y} \tag{7}$$

Where H is the floor height, Δ_y is the yielding lateral deformation of the floor, Δ_{max} represents the maximum lateral deformation of the floor, Δ_u denotes the ultimate displacement at failure (considered here as $\Delta_u = 0.04H$). Classification of the damage state for the first two inter-story drift DI can be made according to [FEMA-356, 2000], as **Table 1** and for Deformation Based DI a classification can be made according to Park and Ang as **Table 2**.

Performance Level	Damage State	DI %
Immediate Occupancy (IO)	No damage	< 0.2
Damage Control (DC)	Minor Damage	< 0.5
Life Safety (LS)	Moderate Damage	< 1.5
Collapse Prevention (CP)	Severe Damage	< 2.5
Collapse	Collapse	> 2.5

Table 1. Interpretation of Inter-Story drift DI.

Damage State	State of the structure	DI
Minor Damage	Serviceable	$0 - 0.2$
Moderate Damage	Repairable	$0.2 - 0.5$
Severe Damage	Irreparable	$0.5 - 1.0$
Collapse Damage	Total loss	> 1.0

Table 2. Interpretation of Deformation Based DI.

1.2 Teaching the Pushover analysis

Teaching the Pushover analysis in Postgraduate program courses of engineering is still a challenging task by many Professors. The inherent dualism of the method similar to the Finite Element method requires a balance between the mathematical theory and physical understanding, and can only be applied through a computer. Even though many textbooks related with the subject contain didactic exercises treating theoretical issues and simple sample problems, they do not contain full sections that may guide the student on how to compute the method by oneself, thus lacking the motivating factor for students to fully appreciate the intimate relationship between the theoretical issues of the method, its physical interpretation and its computer implementation.

On the other hand, there is a tendency by educational practitioners in a course related with this such method to leave the students to operate the method by themselves in any way possible, not really paying attention on the programming skills of each individual. Therefore, those students who lack the good programming knowledge base requirement tend to get frustrated by the so many operations and code required, putting in risk their own learning capabilities.

It has been recommended by some authors [Matti, et al., 2000] that in order to make a computed aided program pedagogical effective such program should be written in such a manner that all the usual subroutines related to the method in question are present; the student should be able to assemble all of such functions and operations so that a calculation is possible, without requiring a large amount of programming effort or computational skills. This way, a much more time-efficient learning environment is enhanced, requiring only for each student to build their own code for every problem to solve, instead of coding and programming the whole computational process with the risk of losing track in the run. Thus, the physical interpretation along with the solution strategy of any problem are continuously connected to the mathematical and numerical treatment of such problem at every moment.

When learning the Pushover analysis method or any other engineering process it is required to review weather or not the calculations are correct, this task is usually observable by the collapse mechanism. It

is thus advantageous for students to be able to visualize the evolution of the deformed structure as it gets closer to the collapse mechanism. Even though software like Microsoft Excel could also provide good mean for the computation of the Pushover method, such package is limited in graphic tools, as they lack of dynamism compared to MatLab graphic functions for instance.

2 Algorithmic process

Algorithm 2.1: Pseudo-code for the Pushover analysis method.

1. Initialize load incremental factor as λ =1.0
2. Apply incremental load factor to the lateral loads to perform a static linear analysis:
    - Check for support conditions at the ends of members and build the corresponding stiffness matrix for each member for their assembly globally
3. Compute mechanic elements of bars and check if any of the members' ends have overpassed their plasticity flexure limit $M \leq Mp$
    - If any end of a member has overpassed such plasticity limit, then change its supports conditions (substitute $''Fixed''$ condition for $''Art''$) and recollect displacements and forces for each DOF. Then apply increment di in the λ factor as $\lambda = \lambda + di$ (recommendation $di = 0.01$)
    - If no member has suffered overpassed such plasticity limit then apply increment di in the λ factor as $\lambda = \lambda + di$ (recommendation $di = 0.01$)
4. Check the terminal criteria of equation (1). If such terminal condition is complied stop the iteration, otherwise go back to step 2
5. Compute Damage Indices

3 Illustrative examples

As following it will be described how to set a whole program to properly use the main $Pushover2DFrames2$ function.

3.1 Structural model 01

$Example\_Pushover\_RCPlane\_Frame\_01$

The structural model frame of **Fig. 4** will be taken as example:

Let us begin by setting the materials for each bar. Given that this example corresponds to a reinforced concrete structure, the Modulus of Elasticity of each bar could be computed in function of their given f'_c as following:

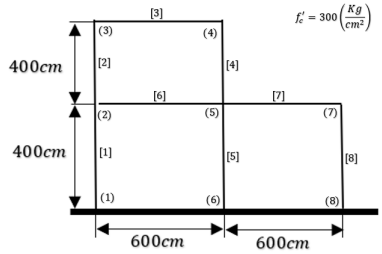

Figure 4. Structural model 01 for testing.

```
%% Materials
% f'c of each element
fpc=[300;
     300;
     250;
     300;
     300;
     250;
     250;
     300];

% Elasticity modulus of each element in function of f'c
E=zeros(nbars,1);
for i=1:nbars
     E(i)=14000*sqrt(fpc(i));
end
```

Now, the geometry data is given, cross-section dimensions and their corresponding geometrical properties, as well as node coordinates:

```
%% Geometry/Topology
% cross-section dimensions of each element (rectangular geometry)
dimensions=[40 40;40 40;30 60;40 40;50 50;30 60;30 50;30 30];

% cross-section area of each element
A=zeros(nbars,1);
for i=1:nbars
     A(i)=dimensions(i,1)*dimensions(i,2);
end

% Cross-section inertia
I=zeros(nbars,1);
for i=1:nbars
     I(i)=1/12*dimensions(i,1)*dimensions(i,2)^3;
end
```

```
% coordinates of each node
coordxy=[0 -100;0 400;0 800;600 800;600 400;600 -100;1200 400;1200 -100];
```

The node connectivity (topology) of the structure must also be given. It suffices to give only the initial and final node of each element, as:

```
%% Topology (connectivity)
ni=[1;2;3;4;5;2;5;7];
nf=[2;3;4;5;6;5;7;8];
```

The boundary conditions are given in the following format of the array bc. It is also necessary to initialize the element support condition vector **supports** which will be modified through out the Pushover analysis as the plastic formations appear at the end of each element:

```
%% Prescribed boudnary conditions [dof, displacement]
bc=[1 0;2 0;3 0;16 0;17 0;18 0;22 0;23 0;24 0];

supports=[1 "Fixed" "Fixed";
          2 "Fixed" "Fixed";
          3 "Fixed" "Fixed";
          4 "Fixed" "Fixed";
          5 "Fixed" "Fixed";
          6 "Fixed" "Fixed";
          7 "Fixed" "Fixed";
          8 "Fixed" "Fixed"];
```

The following data is optional in case it is desired to automate the assignation of loads for beams and columns. However, such loads can be assigned differently by the user:

```
%% Additional data (optional)
type_elem=[1 "Col";
           2 "Col";
           3 "Beam";
           4 "Col";
           5 "Col";
           6 "Beam";
           7 "Beam";
           8 "Col"];

elemcols=[];
elembeams=[];
beams=0;
cols=0;
```

```
for j=1:nbars
    if type_elem(j,2)=="Beam"
        beams=beams+1;
        elembeams=[elembeams,j];
    elseif type_elem(j,2)=="Col"
        cols=cols+1;
        elemcols=[elemcols,j];
    end
end
```

Once it has been established which elements are beams and which are columns, the assignation of loads can be executed as following:

```
%% Loads
beams_LL=[1 100; % Uniformly distributed loads over the beams
          2 100;
          3 100];

% Assignation of distributed loads on beams
qbary=zeros(nbars,2);
for i=1:beams
    qbary(elembeams(i),2)=beams_LL(i,2);
end
```

The initial lateral forces must also be set to start the Pushover method. For such purpose a column vector is set as well as a vector with the corresponding Degrees of Freedom at which such forces act:

```
% Lateral equivalent seismic forces from a modal analysis. The number of
% forces must be equal to the number of floors
seismicForces=[1500; % lower floor
               2000]; % upper floor

% Degrees of freedom over which each seismic force is applied (one for
% each seismic force)
dofSeismicForces=[4 7];
```

Now, assuming that the concrete elements' reinforcement has already been designed, the resistant bending moments at each of their ends must be given as:

```
%% Plastic moments of each element's ends
Mp=[7680000 7680000;
    6490000 6490000;
    8363000 8976940;
    5490000 5490000;
    8680000 8680000;
```

```
    9363000 9976940;
    7363000 7976940;
    5490000 5490000]; %Kg-cm

% Height of each floor
hfloor=[400; 400];
nfloors=length(hfloor);
```

Note that the floors' height must be also be given so that the relative lateral displacements of each floor can be computed.

Finally, the Pushover method is executed for each in-plane direction (left and right) as:

```
%% PUSHOVER IN POSITIVE DIRECTION OF FORCES

[lambdaRight,pdriftDIRight,driftDIRight,defBasedDIRight,maxDispRight,...
 barPlasNodeRight]=Pushover2DFrames2(qbary,A,Mp,E,I,coordxy,ni,nf,...
supports,bc,seismicForces,hfloor,dofSeismicForces,0.01,0.005);

%% PUSHOVER IN NEGATIVE DIRECTION OF FORCES

seismicForces=-seismicForces;

[lambdaLeft,pdriftDILeft,driftDILeft,defBasedDILeft,maxDispLeft,...
 barPlasNodeLeft]=Pushover2DFrames2(qbary,A,Mp,E,I,coordxy,ni,nf,...
supports,bc,seismicForces,hfloor,dofSeismicForces,0.01,0.005);
```

Once both Pushover analysis as executed the critical Collapse Safety Factor is obtained from the minimum of the resultant ones for each analysis direction, which are computed as following from the function output data. The critical of each DI's is determined from minimum of the average for each floor, for each direction to then be classified according to **Table 1** and **Table 2**:

```
%% Final results
SafetyFac=min([max(lambdaRight), max(lambdaLeft)])
pdriftDI=min([sum(pdriftDIRight)/nfloors,sum(pdriftDILeft)/nfloors])
driftDI=min([sum(driftDIRight)/nfloors,sum(driftDILeft)/nfloors])
dbDI=min([sum(defBasedDIRight)/nfloors,sum(defBasedDILeft)/nfloors])

Max_Displacement=max(max(maxDispLeft),max(maxDispRight))
```

3.1.1 Results

The critical Collapse Safety Factor is obtained from the minimum of the resultant ones for each analysis direction $[27.21, 24.39]$ (left and right, respectively), which for this case yields: $[12.88]$. The critical of each DI's is determined from minimum of the average for each floor, for each direction, which for this problem yield $pdrift = 1.78$, $drift = 2.66$, $defbased = 1.0$, for which a Moderate Irreparable Damage state would be estimated according to **Table 1** and **Table 2**.

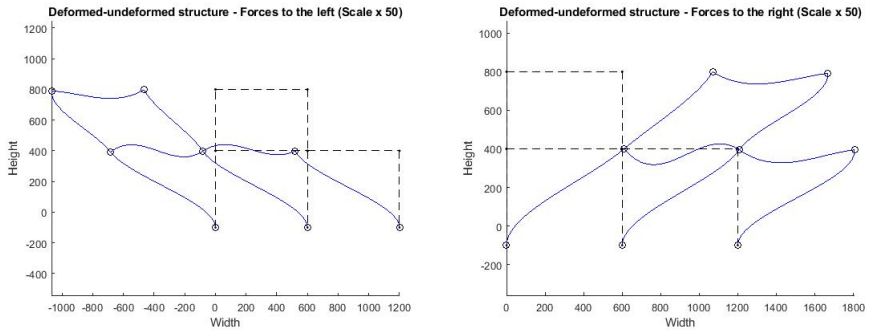

Figure 5. Results obtained from the Pushover analysis for the structural frame 01. (Left) Deformed structure before collapse state due to lateral forces to the left, (Right) Deformed structure before collapse state due to lateral forces to the right.

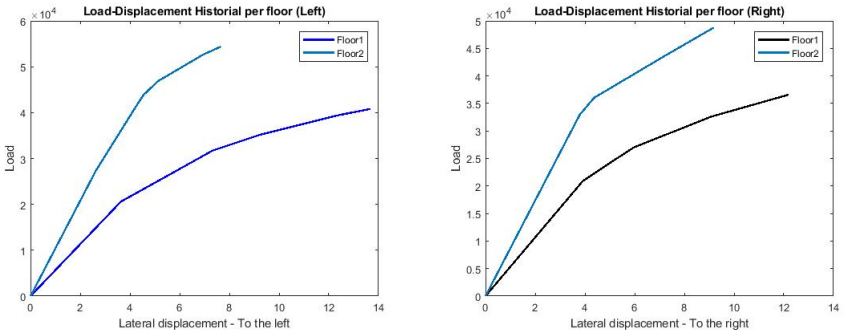

Figure 6. Results obtained from the Pushover analysis for the structural frame 01. (Left) P-Δ of floor 1 and 2 in the left direction of forces, (Right) P-Δ of floor 1 and 2 in the right direction of forces.

References

[Dahlblom et al., 1986] Dahlblom, O., Peterson, A. and Petersson, H. (1986). CALFEM — a program for computer-aided learning of the finite element method, Engineering Computations, Vol. 3 No. 2, pp. 155-160. DOI: https://doi.org/10.1108/eb023653

[Castro, et al., 2004] Castro Trigero, R., Martínez Jimenez, J.M.,Ruiz Sibaja, A., Martínez Jiménez, P., (2004). Implementación de elemento triangular isoparamétrico TDL Shell para flexión de placas y láminas en CALFEM-MatLab, Conference Paper, Anales de Ingeniería Sísmica, Dep. de Mecánica, Universidad de Córdoba, Escuela Politécnica Superior, Córdoba, España

[CALFEM, 2004] CALFEM - A finite element toolbox to MATLAB, Version 3.4, (2004). University of Lund, Structural Mechanics Department, Sweden, 2004. Available: https://www.byggmek.lth.se/english/calfem/

[FEMA 273, 1997] FEMA 273, (1997). NEHRP Guidelines for the seismic rehabilitation of Buildings, Washington, D.C., USA

[FEMA-356, 2000] ASCE-FEMA, (2000). Prestandard and commentary for the seismic rehabilitation of buildings, Reston, Virgina, USA

[Matti, et al., 2000] Ristinmaa, M., Standberg, G., Olsson, K., (2000). CALFEM as a Tool for Teaching University Mechanics, CAL-Laborate International, 5(1). Available: https://www.researchgate.net/publication/268203386_CALFEM_as_a_Tool_for_Teaching_University_Mechanics

[Makhloof et al., 2021] Makhloof, D.A., Ibrahim, A.R., Xiaodan Ren, (2021). Damage Assessment of Reinforced Concrete Structures through Damage Indices: A State-of-the-Art Review, Computer Modeling in Engineering & Sciences, 128,(3). DOI: doi:10.32604/cmes.2021.016882

[Magana & Silva, 2016] Magana, A.J., Silva Couthinho, G., (2016). Modelling and simulation practices for a computational thinking-enabled engineering workforce, Computer Application in Engineering Education, 25,(1). DOI: 10.1002/cae.21779