“Implementation of the Pushover Analysis in MatLab”

Luis Fernando Verduzco Martínez [1][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 Queretaro, 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 **Fig. 2**. Such articulations are simulated through the liberation of the DOF in such plastic formation location and a bending moment equivalent to that for which the plastic formations occurs. This such bending moments forces at each plastic formation location will remain constant

[1]Corresponding author: Autonomous University of Queretaro

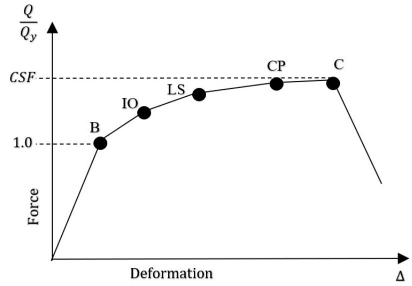

Figure 1. Force-Deformation digram representing the damage states (IO, LS, CP) through the collapse mechanism (C) of ductile structural frame. IO stands for Immediate Occupancy, LS for Life Safety, CP for Collapse Prevention according to [FEMA 273, 1997].

through out the subsequent analysis steps such that the structure's stiffness and force-displacement capacity will be eventually degraded.

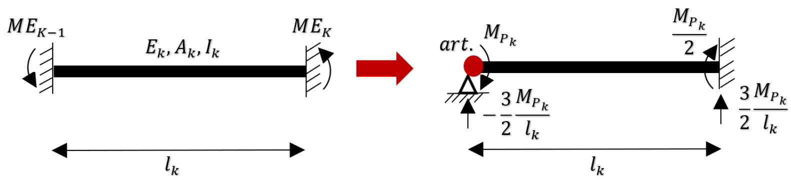

Figure 2. Equivalent structural system transformation when plastic formations at the ends of an element take place.

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), and as it was just mentioned, an equivalent force system should also take place for the latter state condition as (4):

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

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

$$\{f_k\}^T = \begin{bmatrix} 0 & \frac{3M_{p_k}}{2l_k} & M_{p_k} & 0 & -\frac{3M_{p_k}}{2l_k} & \frac{M_{p_k}}{2} \end{bmatrix} \tag{4}$$

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 (5) along with its respective force vector equivalent system (6):

$$[K] = \begin{bmatrix} \frac{EA}{l} & 0 & 0 & -\frac{EA}{l} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{l} & 0 & 0 & \frac{EA}{l} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

$$\{f_k\}^T = \begin{bmatrix} 0 & \frac{2M_{p_k}}{l_k} & M_{p_k} & 0 & -\frac{2M_{p_k}}{l_k} & M_{p_k} \end{bmatrix} \tag{6}$$

1.1 Damage Assessment

In order to evaluate peak dynamic deformation demand of structures at various performance levels, nonlinear 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* (7), *the Plastic Inter-story Drift DI* (8) and *the Deformation Based DI* (9). 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].

$$DI_{drift} = \frac{\Delta_{max}}{H} \tag{7}$$

$$DI_{p-drift} = \frac{\Delta_{max} - \Delta_y}{H} \tag{8}$$

$$DI_{\mu} = \frac{\Delta_{max} - \Delta_y}{\Delta_u - \Delta_y} \tag{9}$$

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. 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 Functions and subroutines

It is expected to build functions or subroutines in such a way that may be of the easiest use for the user. For the functions of this work, the same format used in the CALFEM Toolbox was deployed. In Appendix 1 is presented the MatLab code and the pseudo-code is presented next **Algorithm 2.1**:

Function: ElastoPlasticPushoverPlaneFrames

Purpose: To compute a static non-linear pushover analysis of a plane frame.

Sintax:

$[incfac, pdrift\_DI, drift\_DI, def\_based\_DI, maxDisplacement] = ...$
$ElastoPlasticPushoverPlaneFrames(qbar\_y, a, Mp, nbars, nnodes, ...$
$e, inertia, coordxy, ni, nf, len, co, se, Edof, support, edof, ndof, ...$
$rxbar, rybar, mbar, seismic_forces, h\_floor, dof\_forces, dof\_disp)$

Description:

Input variables:

- a vector containing the area of all elements: size = $[nbars, 1]$
- Mp Plastic Moments for each member: $Mp = [Mpi, Mpj]$ (i) initial node, (j) final node: size = $[nbas, 2]$
- $rxbar$ Reacting horizontal forces by external loads: $[bar, Fxi, Fxj$, size = $[nbars, 3]$
- $rybar$ Reacting vertical forces by external loads; $[bar, Fyi, Fyj$, size = $[nbars, 3]$
- $mbar$ Reacting moments by external loads; $[bar, mi, mj]$, size = $[nbars, 3]$
- e Elasticity modulus of each element, vector of size = $[nbars, 1]$
- $inertia$ in-plane inertia for all elements' cross-section, vector of size = $[nbars, 1]$

- $coordxy$ node coordinates of all nodes, array: $[coordx, coordy]$, size = $[nnodes, 2]$
- ni list of initial nodes of all bars, size = $[nbars, 1]$
- nf list of final nodes of all bars, size = $[nbars, 1]$
- len length of each bar: size = $[nbar, 1]$
- co cos direction of each bar: size = $[nbar, 1]$
- se sin direction of each bar: size = $[nbar, 1]$
- $Edof$ Degrees of freedom of each bar according to its nodes orientation: size = $[nbar, 7]$
- $qbary$ distributed uniform load downward for each beam and only beams: $[bar, load]$, size = $[nbars, 2]$
- $support$ support type at each of the bars' ends. Options: $''Art''$ or $''Fixed''$, size = $[nbar, 2]$
- $edof$ non-restricted Degrees of Freedom:
- $ndof$ number of non-restricted Degrees of Freedom
- $seismic\_forces$ lateral seismic forces per floor: size = $[nfloors, 1]$: in ascendant order (the force of the lower floor is placed first)
- $hfloor$ Height of each floor from bottom to top: size = $[nfloors, 1]$
- dof_forces Degrees of freedom at which the lateral forces are applied (from bottom to top)
- dof_disp dof at which lateral forces are applied in the reduced system of dof reference (edof)

Output variables:

- $incfac$ final incremental load factor at collapse: Collapse Safety Factor (CSF)
- $pdrift\_DI$ Plastic inter-story drift Damage Index per floor, in ascendant order (the DI of the lower floor goes first) size = $[nfloors, 1]$
- $drift\_DI$ Inter-story drift Damage Index per floor, in ascendant order (the DI of the lower floor goes first) size = $[nfloors, 1]$
- def_based_DI Deformation based Damage Index per floor, in ascendant order (the DI of the lower floor goes first) size = $[nfloors, 1]$
- $maxDisplacement$ Max absolute lateral displacement per floor, in ascendant order (the displacement of the lower floor goes first) size = $[nfloors, 1]$

Algoritmo 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 en 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 support condition (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

A following it will be described how to prepare the require data to use the $ElastoPlasticPushoverPlaneFrames$ function.

3.1 Structural model 01

The structural model frame of **Fig. 3** will be taken as example, where [] denote bars and () denote nodes:

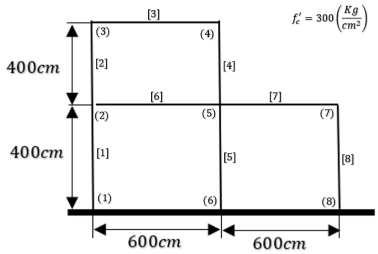

Figure 3. Structural model 01 for testing.

The definition of the elements connection (topology) is established by the array **Edof** as and the node coordinates with **coordxy**:

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

nnodes=length(coordxy(:,1));

ni=[1;2;3;4;5;2;5;7];
nf=[2;3;4;5;6;5;7;8];
nbars=length(ni);

Edof=zeros(nbars,7);
for i=1:nbars
    Edof(i,1)=i;
    Edof(i,2)=ni(i)*3-2;
    Edof(i,3)=ni(i)*3-1;
    Edof(i,4)=ni(i)*3;

    Edof(i,5)=nf(i)*3-2;
    Edof(i,6)=nf(i)*3-1;
    Edof(i,7)=nf(i)*3;

end
```

Material properties and cross-section geometry can inserted as:

```
e=zeros(nbars,1);
for i=1:nbars
    e(i)=14000*sqrt(300);
end

% dimensions = [b,h];
dimensions=[40 40;40 40;30 60;40 40;50 50;30 60;30 50;30 30];

a=zeros(nbars,1);
for i=1:nbars
    a(i)=dimensions(i,1)*dimensions(i,2);
end

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

Boundary conditions can be established as **bc** for prescribed DoF conditions and as **edof** for non-restricted DoF:

```
% prescribed boudnary conditions
bc=[1 0;2 0;3 0;16 0;17 0;18 0;22 0;23 0;24 0];

% habilited nodes
[ndof,edof]=nonRestrcDof(nnodes,bc);
```

It is necessary to initialize the support vector condition **supports** which will be modified through out the Pushover analysis as the plastic formations appear at the end of each element:

```
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"];
```

Loads at the elements are established with the **qbary** array, from which the initial equivalent force vectors **rybar,rxbar,mbar** will be generated **Fig. 4**. This equivalent force vectors are to be modified through out the Pushover assessment as the plastic formations appear **Fig. 2**.

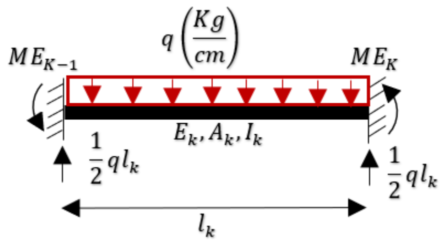

Figure 4. Reactions mechanism due external uniformly distributed loads on a beam element.

```
qbary=[1 0;2 0;3 -100;4 0;5 0;6 -100;7 -100;8 0];

rxbar=zeros(nbars,3);
rybar=zeros(nbars,3);
mbar=zeros(nbars,3);
for i=1:nbars
    rybar(i,2)=qbary(i,2)*l(i)*0.5;
    rybar(i,3)=qbary(i,2)*l(i)*0.5;
```

```
    mbar(i,2)=qbary(i,2)*l(i)^2/12;
    mbar(i,3)=-qbary(i,2)*l(i)^2/12;
end
```

Finally, the Plastic Moment for each bars' ends is specified with **Mp** and the lateral seismic forces as $seismic\_forces$:

```
Mp=[7680000 7680000;
    6490000 6490000;
    8363000 8976940;
    5490000 5490000;
    8680000 8680000;
    9363000 9976940;
    7363000 7976940;
    5490000 5490000]; %Kg-cm

seismic_forces=[1500; % upper floor
                2000]; % lower floor
```

The Pushover analysis should be carried out for both in-plane directions (left and right) as following:

```
%%% IN POSITIVE DIRECTION OF FORCES____________
%______________________________________________________________________________
dof_seismic_forces=[4 7];
dof_disp=[1 4];
hfloor=[400; 400];

[lambda_der,pdrift_DI_der,drift_DI_der,def_based_damage_index_der,...
max_disp_der]=ElastoPlasticPushoverPlaneFrames(qbary,a,Mp,nbars,...
nnodes,e,inertia,coordxy,ni,nf,l,c,s,Edof,supports,edof,ndof,rxbar,...
rybar,mbar,seismic_forces,hfloor,dof_seismic_forces,dof_disp);

%%% IN NEGATIVE DIRECTION OF FORCES____________
%______________________________________________________________________________

seismic_forces=-seismic_forces;

[lambda_izq,pdrift_DI_izq,drift_DI_izq,def_based_damage_index_izq,...
max_disp_izq]=ElastoPlasticPushoverPlaneFrames(qbary,a,Mp,nbars,...
nnodes,e,inertia,coordxy,ni,nf,l,c,s,Edof,supports,edof,ndof,rxbar,...
rybar,mbar,seismic_forces,hfloor,dof_seismic_forces,dof_disp);
```

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**.

```
% Critical CSF:
FS=min([lambda_der, lambda_izq])
% Plastic inter-story drift DI:
pdriftDI=min([sum(pdrift_DI_der)/nfloors,sum(pdrift_DI_izq)/nfloors])
% Inter-story drift DI:
driftDI=min([sum(drift_DI_der)/nfloors,sum(drift_DI_izq)/nfloors])
% Deformation based DI:
dbDI=min([sum(def_based_di_der)/nfloors,sum(def_based_di_izq)/nfloors])
```

3.2 Structural model 02

Analogous to the structural model 01, the input data is formulated for the following topology of **Fig. 5**:

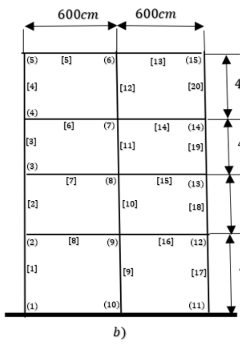

Figure 5. Structural model 01 for testing.

4 Results and Discussion

4.1 Structural model 01

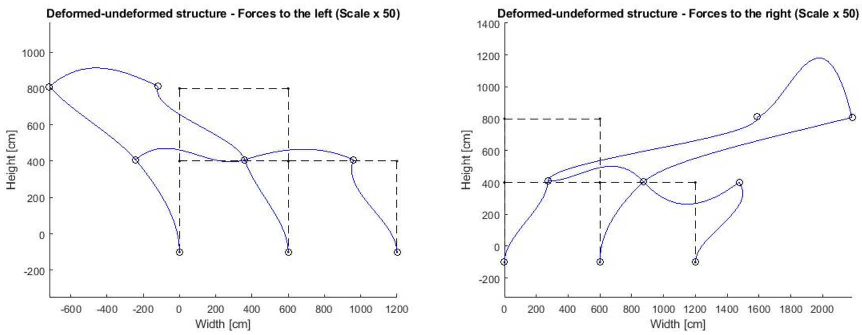

Figure 6. Results obtained from the Pushover analysis for the structural frame 01. (Left) Deformed structure before collapse state due to lateral forces to the left, (Right) Deformed structure before collapse state due to lateral forces to the right.

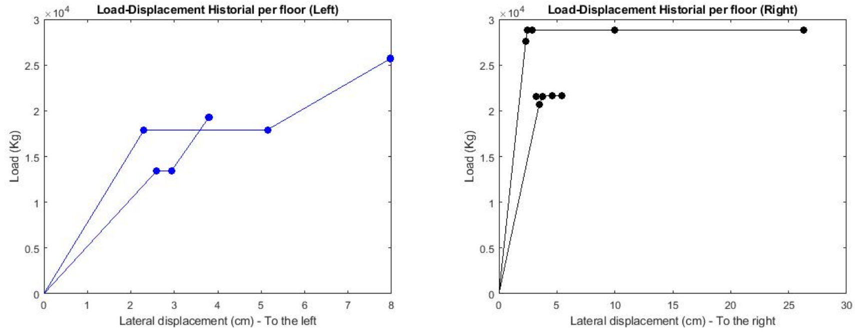

Figure 7. Results obtained from the Pushover analysis for the structural frame 01. (Left) P-Δ of floor 1 and 2 in the left direction of forces, (Right) P-Δ of floor 1 and 2 in the right direction of forces.

The critical Collapse Safety Factor is obtained from the minimum of the resultant ones for each analysis direction $[12.88, 14.43]$, which for this case yields: $[12.88]$ (see following code block). The critical of each DI's is determined from minimum of the average for each floor, for each direction, which for this problem yield $pdrift = 0.86$, $drift = 1.47$, $defbased = 1.0$, for which a Moderate Irreparable Damage state would be estimated according to **Table 1** and **Table 2**.

4.2 Structural model 02

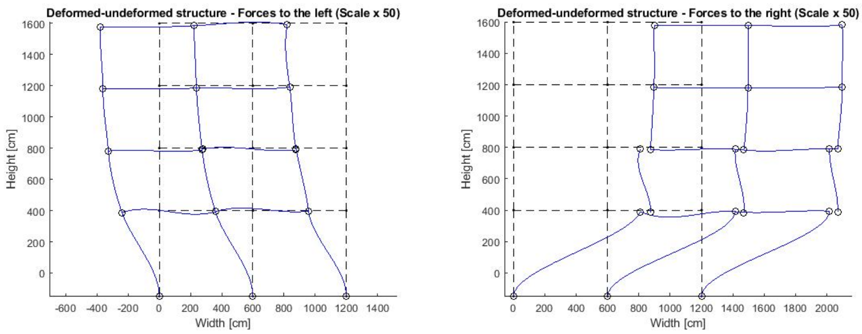

Figure 8. Results obtained from the Pushover analysis for the structural frame 02. (Left) Deformed structure before collapse state due to lateral forces to the left, (Right) Deformed structure before collapse state due to lateral forces to the right.

The critical Collapse Safety Factor is obtained from the minimum of the resultant ones for each analysis direction $[5.55, 5.03]$, which for this case yields: $[5.03]$ (see following code block). The critical of each DI's is determined from minimum of the average for each floor, for each direction, which for this problem yield $DI_{p,drift} = 0.3848$, $DI_{drift} = 0.4739$, $DI_{db} = 1.0$, for which a Minor Irreparable Damage state would be estimated according to **Table 1** and **Table 2**.

5 Funding

This project was supported thanks to CONACyT by means of a scholarship for the Master Program of Science of Structures of the Autonomous University of Queretaro (UAQ).

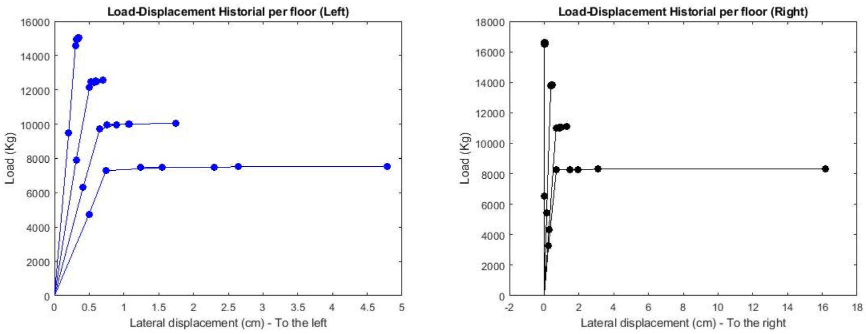

Figure 9. Results obtained from the Pushover analysis for the structural frame 02. (Left) P-Δ of floor 1 and 2 in the left direction of forces, (Right) P-Δ of floor 1 and 2 in the right direction of forces.

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`

[Verduzco, 2022] Luis Fernando Verduzco (2022). Implementation of the Pushover Analysis for Plane Frames (https://www.mathworks.com/matlabcentral/fileexchange/105415-implementation-of-the-pushover-analysis-for-plane-frames), MATLAB Central File Exchange. Retrieved January 18, 2022.