

# **Manual de Apoyo al Análisis de Ensayos Cíclicos de Muros Hormigón Armado**

*Script de Matlab Analisis<sub>D</sub>ata<sub>E</sub>xperimental.m*

GABRIEL FOLLET

*Pontificia Universidad Católica de Chile  
Departamento de Ingeniería Estructural y Geotécnica  
gabriel.follet@uc.cl  
v3-diciembre 2022*

## ÍNDICE GENERAL

---

<b>Índice general</b>	<b>1</b>
<b>Índice de figuras</b>	<b>3</b>
<b>1 Introducción</b>	<b>4</b>
<b>2 Carga de Data Experimental</b>	<b>5</b>
<b>3 Outputs</b>	<b>6</b>
<b>4 Perfiles y Opciones manuales</b>	<b>7</b>
<b>5 Data.Procesada.Data</b>	<b>8</b>
5.1. DataProcesada.Data.RawData . . . . .	8
5.2. DataProcesada.Data.ProcessedData . . . . .	8
<b>6 Data.Procesada.Ciclos</b>	<b>10</b>
6.1. Data.Procesada.Ciclos.DataCiclos . . . . .	10
6.2. Data.Procesada.Ciclos.puntoscriticos . . . . .	10
6.3. Data.Procesada.Ciclos.Protocolo . . . . .	10
<b>7 DataProcesada.Deriva</b>	<b>11</b>
7.1. DataProcesada.Deriva.MaxDeriva . . . . .	11
7.2. DataProcesada.Deriva.MinDeriva . . . . .	11
<b>8 DataProcesada.Fuerza</b>	<b>12</b>
8.1. DataProcesada.Fuerza.MaxFuerza . . . . .	12
8.2. DataProcesada.Fuerza.MinFuerz . . . . .	12
<b>9 DataProcesada.Rigidez</b>	<b>13</b>
9.1. DataProcesada.Rigidez.Carga . . . . .	14
9.2. DataProcesada.Rigidez.Descarga . . . . .	14
9.3. DataProcesada.Rigidez.Secante . . . . .	15
9.4. Rigideces en último ciclo . . . . .	16
<b>10 DataProcesada.Energia</b>	<b>17</b>
10.1. DataProcesada.Energia.EnergiaDisipada . . . . .	17
10.2. DataProcesada.Energia.EnergiaDisipadaAcumulada . . . . .	17
10.3. DataProcesada.Energia.EnergiaDisipadaAcumuladaNormalizada . . . . .	17
<b>11 DataProcesada.Info</b>	<b>20</b>
11.1. DataProcesada.Info.Fuente . . . . .	20
11.2. DataProcesada.Info.ShearSpan . . . . .	20
11.3. DataProcesada.Info.Unidades . . . . .	20
11.4. DataProcesada.Info.ID . . . . .	20
<b>12 DataProcesada.PerdidaRigidez</b>	<b>21</b>
12.1. Normalización . . . . .	21
12.2. DataProcesada.PerdidaRigidez.Carga . . . . .	22
12.3. DataProcesada.PerdidaRigidez.Descarga . . . . .	22

12.4. DataProcesada.PerdidaRigidez.Secante . . . . .	22
<b>13 DataProcesada.PerdidaFuerza</b>	<b>23</b>
13.1. DataProcesada.PerdidaFuerza.FuerzaMaxima . . . . .	23
13.2. DataProcesada.PerdidaFuerza.FuerzaMinima . . . . .	23
<b>14 DataProcesada.Backbone</b>	<b>24</b>
14.1. DataProcesada.Backbone.Backbone . . . . .	24
14.2. DataProcesada.Backbone.ModeloBilineal . . . . .	24
14.3. DataProcesada.Backbone.Falla . . . . .	25
14.4. DataProcesada.Backbone.Ductilidad . . . . .	27
<b>15 DataProcesada.Otros</b>	<b>28</b>
15.1. <i>DataProcesada.Otros.Outlier</i> . . . . .	28
15.2. <i>DataProcesada.Otros.PuntosNegInicial</i> . . . . .	28
15.3. <i>DataProcesada.Otros.CiclosMuyCortos</i> . . . . .	28
15.4. <i>DataProcesada.Otros.Comentarios</i> . . . . .	28
<b>16 Anexo A. Funciones Auxiliares</b>	<b>29</b>

## ÍNDICE DE FIGURAS

---

1.	Carga de data en Script . . . . .	5
2.	Impresión de Resumen en la consola . . . . .	6
3.	Gráficas Validación Visual . . . . .	6
4.	struct <i>DataProcesada</i> con sus atributos . . . . .	6
5.	Algunas opciones dentro del Script . . . . .	7
6.	struct <i>DataProcesada.Data</i> con sus atributos . . . . .	8
7.	Resultado de interpolación de puntos para un ciclo . . . . .	9
8.	struct <i>DataProcesada.Ciclos</i> con sus atributos . . . . .	10
9.	struct <i>DataProcesada.Deriva</i> con sus atributos . . . . .	11
10.	struct <i>DataProcesada.Fuerza</i> con sus atributos . . . . .	12
11.	struct <i>DataProcesada.Rigidez</i> con sus atributos . . . . .	13
12.	Estimación de rigideces en un ciclo . . . . .	15
13.	struct <i>DataProcesada.Energia</i> con sus atributos . . . . .	17
14.	Energía disipada en un ciclo . . . . .	17
15.	Energía de normalización con <i>ENorm</i> =1 . . . . .	18
16.	Energía de normalización con <i>ENorm</i> =2 . . . . .	18
17.	struct <i>DataProcesada.Info</i> con sus atributos . . . . .	20
18.	struct <i>DataProcesada.PerdidaRigidez</i> con sus atributos . . . . .	21
19.	Comparación distintos valores de <i>Q</i> . . . . .	21
20.	struct <i>DataProcesada.PerdidaFuerza</i> con sus atributos . . . . .	23
21.	struct <i>DataProcesada.Backbone</i> con sus atributos . . . . .	24
22.	Ajuste modelo bilineal 1 . . . . .	25
23.	Ajuste modelo bilineal 2 . . . . .	25
24.	Puntos de falla . . . . .	26
25.	Resultados de detección de <i>outliers</i> . . . . .	29
26.	Validación visual de backbone y bilineal . . . . .	30
27.	Validación visual de ciclos identificados . . . . .	30
28.	Validación visual de ciclos y rigideces. . . . .	31
29.	Validación visual de detección de ciclo . . . . .	31
30.	Pérdida de rigidez en función de deriva . . . . .	32
31.	Pérdida de rigidez en función de deriva . . . . .	32

## INTRODUCCIÓN

---

Este manual funciona como un apoyo al script *Analisis\_Data\_Experimental.m* desarrollado para analizar ensayos cíclicos reversibles a deformación controlada.

El script ajusta un backbone y un modelo bilineal a la data experimental. También calcula múltiples rigideces, energía disipada y ductilidad, entre otros parámetros. La data experimental del ensayo debe de estar “relativamente limpia”, en específico, **no** debe que poseer:

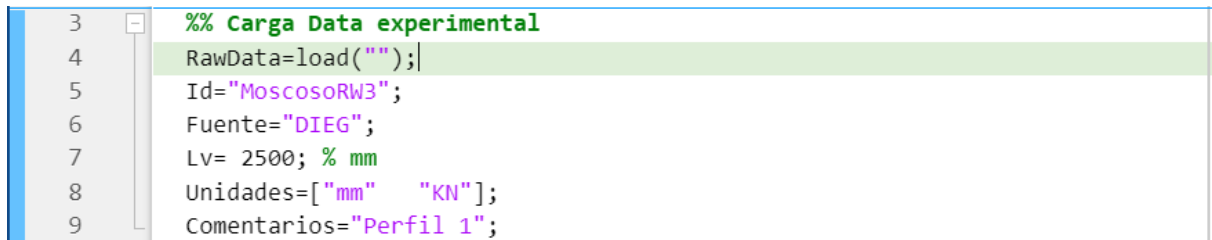
- Pausas iniciales.
- Pausas finales.
- Pausas intermedias donde la deformación oscila entorno al cero.

El script *Analisis\_Data\_Experimental.m* y las funciones auxiliares del Anexo A, se encuentran disponibles en [https://github.com/GabrielFollet/IPre\\_ICE-2985](https://github.com/GabrielFollet/IPre_ICE-2985)

## CARGA DE DATA EXPERIMENTAL

---

Es necesario cargar dentro del Script 6 variables. En la figura 1 se muestra a sección de carga de data dentro del Script.

A screenshot of a script editor window. On the left, a vertical line of numbers 3 through 9 indicates line numbers. The script text is as follows: Line 3: %% Carga Data experimental; Line 4: RawData=load("");; Line 5: Id="MoscosoRW3"; Line 6: Fuente="DIEG"; Line 7: Lv= 2500; % mm; Line 8: Unidades=["mm" "KN"]; Line 9: Comentarios="Perfil 1";. The text is color-coded: comments are green, strings are pink, and other identifiers are black.

```
3 %% Carga Data experimental
4 RawData=load("");
5 Id="MoscosoRW3";
6 Fuente="DIEG";
7 Lv= 2500; % mm
8 Unidades=["mm" "KN"];
9 Comentarios="Perfil 1";
```

Figura 1: Carga de data en Script

El formato y significado de cada una de estas variables es :

- **Raw Data** : Corresponde a la data experimental del ensayo. La estructura de esta data debe de ser siguiente
  - **RawData(:,1)** = Vector columna con los data de la deformación.
  - **RawData(:,2)** = Vector columna con la data de la fuerza.
- **Id**: Corresponde a un string array con la identificación del muro ensayado.
- **Fuente**: Corresponde a un string con la fuente de la data experimental.
- **Lv**: Corresponde a un float con el ShearSpan del muro, es decir con al altura desde la base del muro al actuador.
- **Unidades** : Corresponde a un vector fila. El primer elemento corresponde a un string con la unidad de la deformación y el segundo elemento a un string con la unidad de fuerza. Es importante recalcar que las unidad de la deformación debe de ser consistente con la unidad de *Lv*.
- **Comentarios**: Esta variables es un string que se utiliza parar almacenar comentarios relacionados a la data experimental.

## OUTPUTS

La rutina tiene 3 *outputs*. El primero, consiste en la impresión en la consola de un resumen de lo realizado por el Script, el segundo corresponde a gráficas útiles para validar gráficamente lo realizado por el Script, finalmente un structure array llamado *DataProcesada* compuesto por 11 atributos. En la figuras 2, 3 y 4 se muestra un ejemplo de los *outputs* del script.

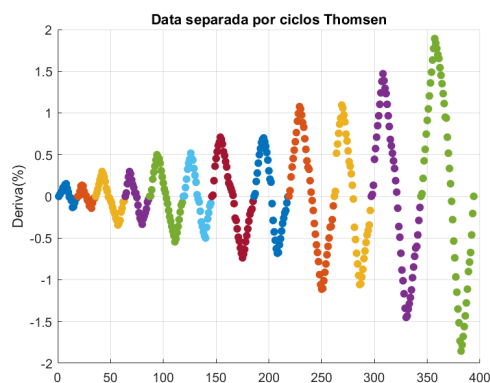
```
-----
Ensayo  Thomsen
-----

Unidades: mm , KN
ShearSpan:4520 mm

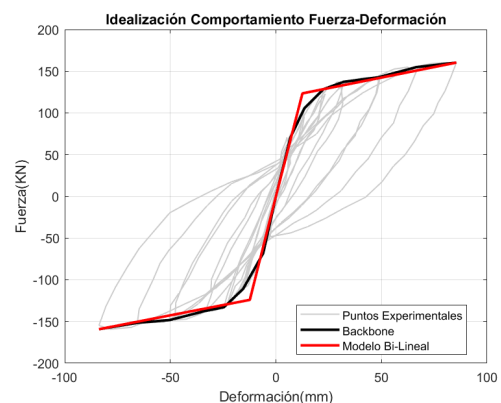
Historia de Manipulación/Alteración:
  Se identificaron y eliminaron 0 puntos muestrales irregulares
  Se eliminaron 1 puntos muestrales iniciales con fuerza o deformación negativa
  Se agregó punto inicial (0,0)
  Se agregaron 11 puntos tipo (0,f) interpolados linealmente para definir limites de ciclos
  Se eliminaron 2 ciclos por no cumplir con los requisitos mínimos

Se identificó el siguiente protocolo de ensayo:
```

Figura 2: Impresión de Resumen en la consola



(a) Validación visual: Detección de ciclo



(b) Validación Visual: Modelo Ajustado

Figura 3: Gráficas Validación Visual

DataProcesada	
1x1 struct with 11 fields	
Field	Value
Data	1x1 struct
Ciclos	1x1 struct
Deriva	1x1 struct
Fuerza	1x1 struct
Rigidez	1x1 struct
Energia	1x1 struct
Info	1x1 struct
PerdidaRigidez	1x1 struct
PerdidaFuerza	1x1 struct
Backbone	1x1 struct
Otros	1x1 struct

Figura 4: struct *DataProcesada* con sus atributos

En los siguientes capítulos se explica el contenido del struct *DataProcesada*

## PERFILES Y OPCIONES MANUALES

El script utiliza múltiples variables para ajustar el funcionamiento. Definiéndose 11 variables, que pueden ser modificadas en la sección *Perfiles y Parámetros opcionales*. Para facilitar el funcionamiento se definieron 2 perfiles que establecen los valores de estas variables, como se observa en la figura 5. A continuación se mencionan brevemente el significado de cada perfil.

- Perfil=1

Perfil que corresponde a ajustar un Backbone considerando el punto de deformación máxima del primer ciclo a cada nivel de deformación, y ocupa este backbone para ajustar un modelo bi-lineal con rigidez post-fluencia.

- Perfil=2

Perfil que corresponde a ajustar un Backbone considerando el punto de deformación máxima del primer ciclo a cada nivel de deformación, y ocupar este backbone para ajustar un modelo bi-lineal elasto-plástico.

A continuación se muestra la sección del código donde están todas las variables a modificar, con una breve descripción, en el presente manual se explicarán con mayor detalle y se marcarán con *color azul*, para facilitar la comprensión.

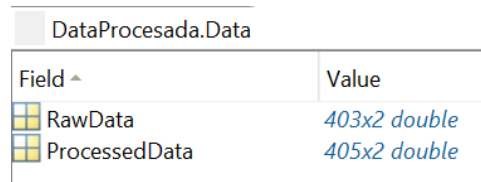
```
10 %% Perfiles y parámetros opcionales
11
12 %Perfiles con posibles combinaciones de los parametros
13 Perfil=1;
14
15 %Pre-Procesamiento
16 window= 20; % largo de ventana para eliminación de outliers con metodo mov-median
17
18 %BackBone
19 BackboneOption=1; % Modelo de backbone a ajustar
20 B_Tolerancia_intermedia=0.95; % Tolerancia para determinar puntos de backbone no finales
21 B_Tolerancia_final=0.97; % Tolerancia para punto final , criterio falla 1
22 B_tolerancia_Perdida_F=0.10; % Tolerancia para pérdida de fuerza en grupo -> muro falla
23 B_tolerancia_Perdida_F_final=0.99; % Tolerancia para perdida de fuerza en último ciclo -> muro falla
24
25 %Bi-Lineal
26 BilinearOption=1; % Modelo bi-lineal a ajustar
27 EnergyTolerance=0.05; % tolerancia para el error maximo en areas bi-lineal
28
29 %Normalización
30 KNormOption=2; % metodo de normalización de rigidez
31 Q=0.6; % Valor que define punto en envolvente que define la rigidez de referencia
32
33 %Ductilidad
34 DuctilityOption=2; % determina is se considera punto de falla como def última o ultimo de bi-lineal
35
36 %Detección de Ciclos y Protocoloes
37 Alpha=0.1; % mínimo de puntos por ciclo(% de segundo ciclo con menos puntos)
38 MinPointsCicle=6; % minimo de puntos por ciclo
39 CicleTolerance=0.15; % diferencia maxima entre ciclos a igual deformación para que se consideren parte del mismo grupo
40 PartialCicleTolerance=0.6; % tolerancia para decidir si un ciclo parcial se usa para calcular kc kd y ks
41 FinalPointTolerance=0.4; % el ancho de la descarga en parte negativa tiene que ser al menos el %d del ancho de la parte negativa
42 MidPointTolerance=0.6; % el ancho de la descarga en parte positiva tiene que ser al menos el %d del ancho de la parte positivo
```

Figura 5: Algunas opciones dentro del Script



## DATA.PROCESADA.DATA

El atributo *Data.Procesada.Data* es a su vez un struct con 2 atributos, según lo que se muestra en la figura 6.



Field ^	Value
RawData	403x2 double
ProcessedData	405x2 double

Figura 6: struct *DataProcesada.Data* con sus atributos

A continuación se explican el contenido de cada uno de los atributos

### 5.1. DATA.PROCESADA.DATA.RAWDATA

Corresponde a un array con la data cargada originalmente

### 5.2. DATA.PROCESADA.DATA.PROCESSEDDATA

Corresponde a un array con la data procesada. El procesamiento consisten en:

- Eliminación de *Outliers*
- Eliminación de puntos iniciales
- Interpolación de puntos

#### ELIMINACIÓN DE OUTLIERS

La detección de *outlier* se realiza con la función interna de Matlab *rmoutliers*, particularmente se decidió utilizar el método *movmedian*, considerando una ventana de tamaño *window* para el cálculo de la mediana móvil. Queda a discreción del usuario la utilización de este método o utilizar otros de los que Matlab posee.

#### ELIMINACIÓN DE PUNTOS INICIALES

Si bien se presupone que la data experimental está “relativamente limpia”, se realiza una pequeña limpieza de la data que consiste en :

- Eliminar todos los puntos iniciales que tiene deformación negativa o fuerza negativa.
- Insertar el punto (0, 0) en la primera posición.

#### INTERPOLACIÓN DE PUNTOS

Se define que un ciclo comienza cuando la deformación pasa de negativa a positiva. Es así, que el Script interpola linealmente entre estos dos puntos agregando el punto (0,  $f$ ) el que marca el fin del ciclo anterior y el inicio de un nuevo ciclo. También, se interpolan los puntos ( $d^+$ , 0) y ( $d^-$ , 0), que definen la parte positiva y negativa de cada ciclo. A continuación se presenta un ejemplo del resultado de estas interpolaciones

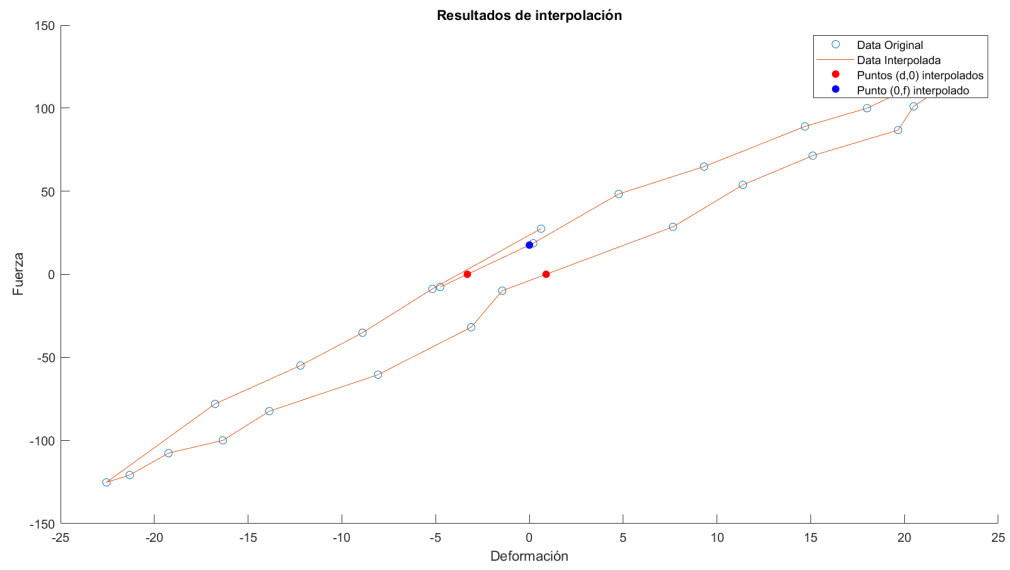
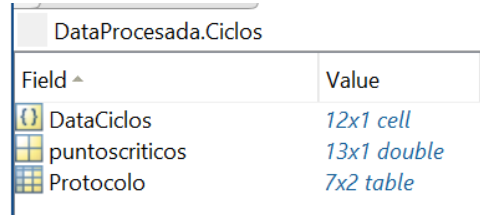


Figura 7: Resultado de interpolación de puntos para un ciclo

## DATA.PROCESADA.CICLOS

El atributo *Data.Procesada.Ciclos* es a su vez un struct con 3 atributos, según lo que se muestra en la siguiente figura.



Field ^	Value
DataCiclos	12x1 cell
puntoscriticos	13x1 double
Protocolo	7x2 table

Figura 8: struct *DataProcesada.Ciclos* con sus atributos

A continuación, se explica el contenido de cada uno de los atributos

### 6.1. DATA.PROCESADA.CICLOS.DATACICLOS

Corresponde un cell array, donde cada elemento es un array con la data experimental de los puntos de los ciclos detectados.

Si algún ciclo presenta insuficiente puntos muestrales, el ciclo, y todos los puntos asociados son eliminados de la data procesada. La cantidad mínima de puntos queda determinado por

$$\max(\text{MinPointCicle}, \text{Alpha} * \text{length}(\text{DataCiclos1}))$$

### 6.2. DATA.PROCESADA.CICLOS.PUNTOSCRITICOS

Corresponde a un vector con los puntos críticos. Se define como punto crítico a un punto donde la deformación es exactamente igual a cero. Los que corresponden a puntos que originalmente tenían deformación igual a 0 y los puntos recientemente interpolados.

### 6.3. DATA.PROCESADA.CICLOS.PROTOCOLO

Corresponde a una tabla con el protocolo de carga identificado. Se ajustó el protocolo considerando la (semi)amplitud y el número de ciclos a cada amplitud.

La semi-amplitud de cada ciclo se determinó a partir de la siguiente convención

- Ciclo Inicial  
La semi-amplitud es el mayor valor entre la deformación máxima y mínima
- Ciclos intermedios  
El valor de la semi-amplitud es el promedio entre la deformación máxima y mínima.
- Ciclo Final  
La semi-amplitud es el mayor valor entre la deformación máxima y mínima

Esta parte de Script se define una variable llamada *deformacion* que guarda el valor de la semi-amplitud a la cual comparar, si la semi-amplitud de un ciclo subsecuente es igual a la semi-amplitud a comparar  $\pm$ , *CicleTolerance*, se considera que este ciclo pertenece al grupo de ciclos de igual deformación. Este valor se actualiza, con el valor del promedio de todos los ciclos que pertenecen al grupo.

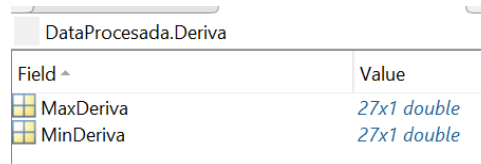
El intervalo que define si un ciclo pertenece a un grupo, queda definido entonces por

$$(\text{deformacion} * (1 - \text{CicleTolerance}), \text{deformacion} * (1 + \text{CicleTolerance}))$$

## DATAPROCESADA.DERIVA

---

El atributo *DataProcesada.Deriva* corresponde a su vez a un struct con dos atributos como se muestra en la siguiente figura .



DataProcesada.Deriva	
Field ^	Value
MaxDeriva	27x1 double
MinDeriva	27x1 double

Figura 9: struct *DataProcesada.Deriva* con sus atributos

### 7.1. DATAPROCESADA.DERIVA.MAXDERIVA

Corresponde a un array con la deriva máxima alcanzada en cada ciclo

### 7.2. DATAPROCESADA.DERIVA.MINDERIVA

Corresponde a un array con la deriva mínima alcanzada en cada ciclo

## DATAPROCESADA.FUERZA

---

El atributo *DataProcesada.Fuerza* corresponde a su vez a un struct con dos atributos como se muestra en la siguiente figura .

DataProcesada.Fuerza	
Field ▲	Value
FuerzaMax	12x1 double
FuerzaMin	12x1 double

Figura 10: struct *DataProcesada.Fuerza* con sus atributos

### 8.1. DATAPROCESADA.FUERZA.MAXFUERZA

Corresponde a un array con la fuerza máxima alcanzada en cada ciclo

### 8.2. DATAPROCESADA.FUERZA.MINFUERZ

Corresponde a un array con la fuerza mínima alcanzada en cada ciclo

## DATAPROCESADA.RIGIDEZ

El atributo *DataProcesada.Rigidez* corresponde a su vez a un struct con tres atributos como se muestra en la siguiente figura.

DataProcesada.Rigidez	
Field ▲	Value
Carga	4x12 double
Descarga	4x12 double
Secante	2x12 double

Figura 11: struct *DataProcesada.Rigidez* con sus atributos

Se consideraron 2 maneras de estimar la rigidez dentro de un ciclo

### MÉTODO 1

Considera que la rigidez queda determinado por el punto de fuerza máxima. Se calculan las siguientes rigideces con este método

- Rigidez positiva de Carga ( $K_{c1}$ )  
Corresponde a la rigidez determinado por los puntos

$$(D_i, F = 0) \quad || \quad (D, F_{max})$$

Donde  $D_i$  corresponde al punto donde la fuerza es nula en carga.

- Rigidez Positiva de Descarga ( $K_{d1}$ )  
Corresponde a la rigidez determinado por los puntos

$$(D, F_{max}) \quad || \quad (D_j, F = 0)$$

Donde  $D_j$  corresponde al punto donde la fuerza es nula en descarga.

- Rigidez Negativa de Carga ( $K_{c1}^-$ )  
Corresponde a la rigidez determinado por los puntos

$$(D_j, F = 0) \quad || \quad (D, F_{min})$$

Donde  $D_j$  corresponde al punto donde la fuerza es nula en carga.

- Rigidez Negativa de Descarga ( $K_{d1}^-$ )  
Corresponde a la rigidez determinado por los puntos

$$(D, F_{min}) \quad || \quad (D_i, F = 0)$$

Donde  $D_i$  corresponde al punto donde la fuerza es nula en descarga.

- Rigidez Secante ( $K_{s1}$ ) Corresponde a la rigidez determinado por los puntos

$$(D, F_{min}) \quad || \quad (D, F_{max})$$

## MÉTODO 2

Considera que la rigidez queda determinado por el punto de deformación máxima.  
Se calculan las siguientes rigideces con este método

- Rigidez positiva de Carga ( $K_{c2}$ )  
Corresponde a la rigidez determinado por los puntos

$$(D_i, F = 0) \quad || \quad (D_{max}, F)$$

Donde  $D_i$  corresponde al punto donde la fuerza es nula en carga.

- Rigidez Positiva de Descarga ( $K_{d2}$ )  
Corresponde a la rigidez determinado por los puntos

$$(D_{max}, F) \quad || \quad (D_j, F = 0)$$

Donde  $D_j$  corresponde al punto donde la fuerza es nula en descarga.

- Rigidez Negativa de Carga ( $K_{c2}^-$ )  
Corresponde a la rigidez determinado por los puntos

$$(D_j, F = 0) \quad || \quad (D_{min}, F)$$

Donde  $D_j$  corresponde al punto donde la fuerza es nula en carga.

- Rigidez Negativa de Descarga ( $K_{d2}^-$ )  
Corresponde a la rigidez determinado por los puntos

$$(D_{min}, F) \quad || \quad (D_i, F = 0)$$

Donde  $D_i$  corresponde al punto donde la fuerza es nula en descarga.

- Rigidez Secante ( $K_{s2}$ ) Corresponde a la rigidez determinado por los puntos

$$(D_{min}, F) \quad || \quad (D_{max}, F)$$

A continuación se muestra gráficamente las rigideces calculadas en cada ciclo.

### 9.1. DATAPROCESADA.RIGIDEZ.CARGA

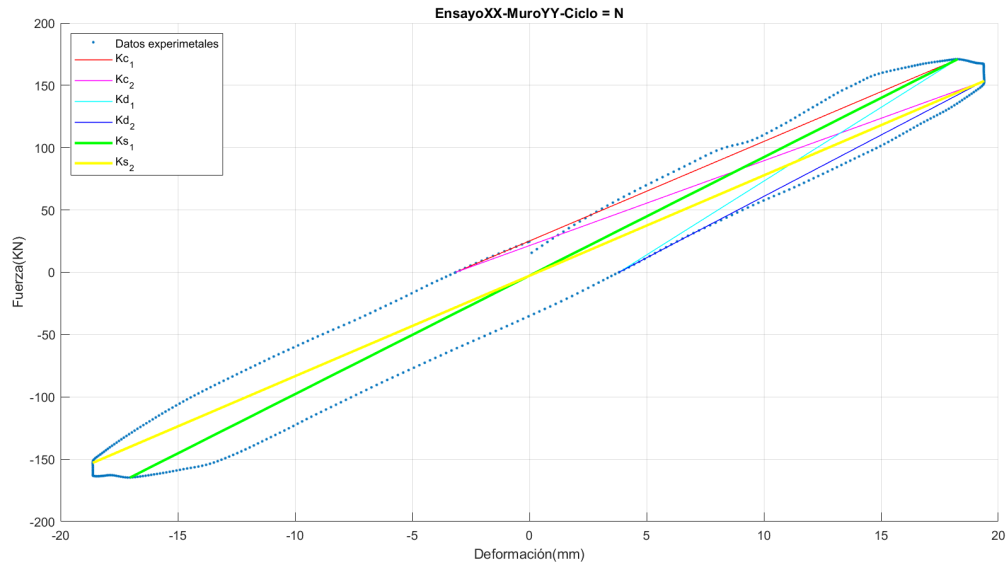
Corresponde a un array donde:

- La primera fila es la rigidez de carga por el método 1 en la parte positiva en cada ciclo.
- La segunda fila es la rigidez de carga por el método 2 en la parte positiva en cada ciclo.
- La tercera fila es la rigidez de carga por el método 1 en la parte negativa en cada ciclo.
- La cuarta fila es la rigidez de carga por el método 2 en la parte negativa en cada ciclo.

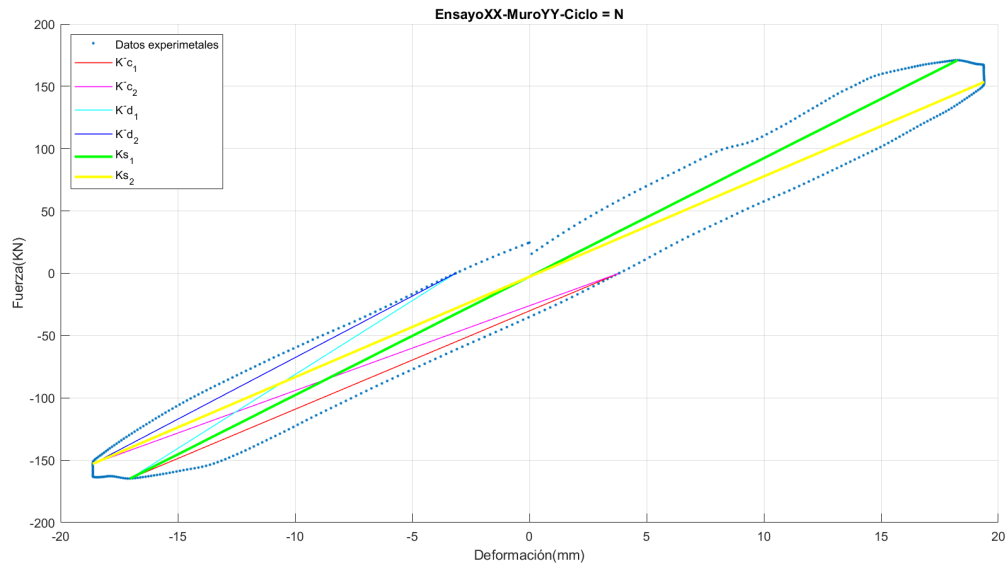
### 9.2. DATAPROCESADA.RIGIDEZ.DESCARGA

Corresponde a un array donde:

- La primera fila es la rigidez de descarga por el método 1 en la parte positiva en cada ciclo.
- La segunda fila es la rigidez de descarga por el método 2 en la parte positiva en cada ciclo.
- La tercera fila es la rigidez de descarga por el método 1 en la parte negativa en cada ciclo.
- La cuarta fila es la rigidez de descarga por el método 2 en la parte negativa en cada ciclo.



(a) Rigideces parte positiva



(b) Rigideces parte negativa

Figura 12: Estimación de rigideces en un ciclo

### 9.3. DATAPROCESADA.RIGIDEZ.SECANTE

Corresponde a un array donde:

- La primera fila es la rigidez de secante por el método 1.
- La segunda fila es la rigidez de secante por el método 2.



#### 9.4. RIGIDECES EN ÚLTIMO CICLO

Las convenciones explicada anteriormente para estimar la rigidez de un ciclo, pueden no ser adecuadas si el ciclo final no se completo, es decir, si el ultimo punto experimental no es (0, 0). De esta manera se consideró por defecto que la rigidez del último ciclo ( para todas las posibles tipos ) es nula, y se establecieron condiciones para verificar que las convenciones de rigidez definidas anteriormente sean aplicables y representativos del ciclo. Las condiciones para cada "tipo de rigidez" son

- .RIGIDEZ.CARGA(1,:)

Se considera que el método es aplicable si

$$\text{PartialCycleTolerance} * \max f(end - 1) < \max f(end)$$

- .RIGIDEZ.CARGA(2,:)

Se considera que el método es aplicable si

$$\text{PartialCycleTolerance} * \max def(end - 1) < \max def(end)$$

- .RIGIDEZ.CARGA(3,:) y .RIGIDEZ.SECANTE(1,:)

Se considera que el método es aplicable si existen puntos muestrales en la parte negativa de las histéresis y si

$$\text{PartialCycleTolerance} * \min f(end - 1) > \min f(end)$$

- .RIGIDEZ.CARGA(4,:) y .RIGIDEZ.SECANTE(2,:)

Se considera que el método es aplicable si existen puntos muestrales en la parte negativa de las histéresis y si

$$\text{PartialCycleTolerance} * \min def(end - 1) > \min def(end)$$

- .RIGIDEZ.DISCARGA(1,:) y .RIGIDEZ.DISCARGA(2,:)

Se considera que el método es aplicable si

Existe un punto con coordenadas (d, 0), punto que solo existe según lo mencionado en Interpolación de Puntos

O, si "la deformación del ultimo punto experimental es menor que una fracción de la deformación máxima del ciclo", es decir

$$\text{Ciclos}\{end\}(end, 1) > \text{MidPointTolerance} * \max def(end)$$

- .RIGIDEZ.DISCARGA(3,:) y .RIGIDEZ.DISCARGA(4,:)

Se consideran que lo método anterior aplican si el valor absoluto de la deformación del ultimo punto experimental es menor que una fracción del valor absoluto de la deformación mínima del ciclo", es decir

$$\text{Ciclos}\{end\}(end, 1) > \text{FinalPointTolerance} * \text{abs}(\min def(end))$$

## DATAPROCESADA.ENERGIA

El atributo *DataProcesada.Energia* corresponde a su vez a un struct con tres atributos como se muestra en la siguiente figura .

DataProcesada.Energia	
Field ^	Value
EnergiaDisipada	12x1 double
EnergiaDisipadaAcumulada	12x1 double
EnergiaDisipadaAcumuladaNormalizada	1x12 double

Figura 13: struct *DataProcesada.Energia* con sus atributos

### 10.1. DATAPROCESADA.ENERGIA.ENERGIADISIPADA

Corresponde a un vector con la energía disipada en cada ciclo. Se calculó la energía como el área encerrada por cada ciclo como se observa en la figura 14 .

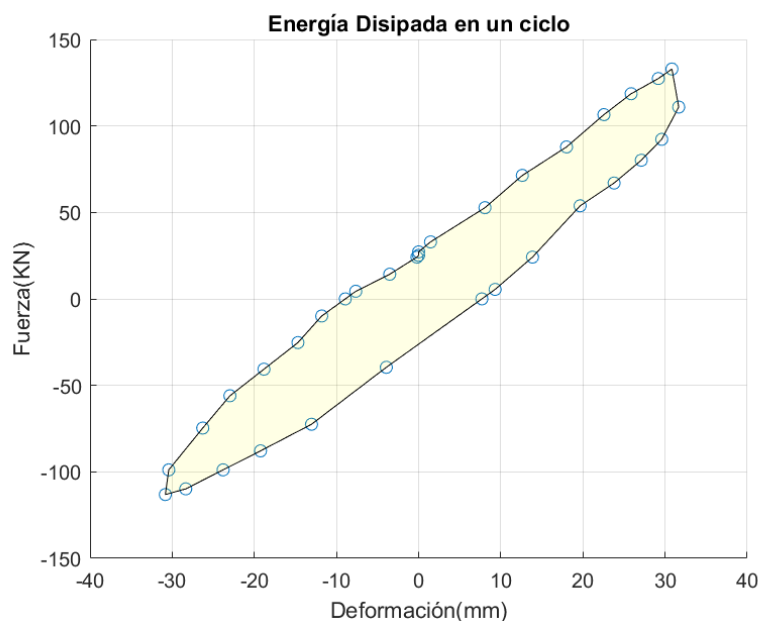


Figura 14: Energía disipada en un ciclo

### 10.2. DATAPROCESADA.ENERGIA.ENERGIADISIPADAACUMULADA

Corresponde a un vector con la energía disipada acumulada hasta cada ciclo.

### 10.3. DATAPROCESADA.ENERGIA.ENERGIADISIPADAACUMULADANORMALIZADA

Corresponde a un vector con la energía disipada acumulada hasta el fin de cada ciclo, normalizado con la energía disipada por el modelo bi-lineal ajustado, la variable *ENorm*, define que estimación de la energía disipada se utiliza en la normalización. En particular, si

- *ENorm*=1, Se considera que el área encerrada toma en cuenta una descarga lineal, como se muestra en la figura 16

- $ENorm=2$ , Se considera que el área encerrada no considera una descarga como se muestra en la figura??

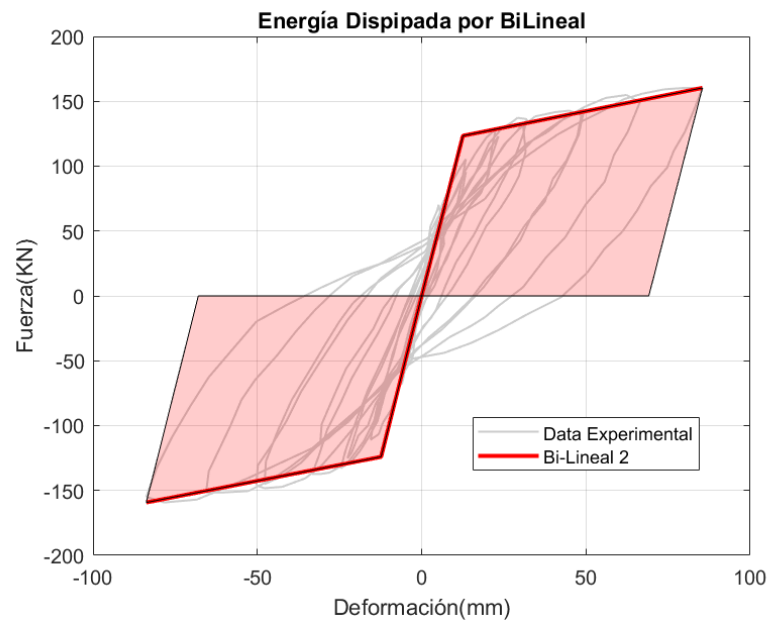


Figura 15: Energía de normalización con  $ENorm=1$

..

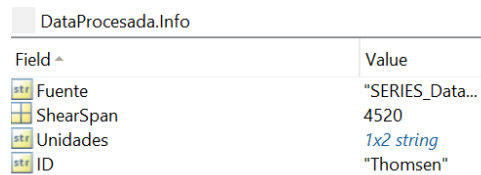
Figura 16: Energía de normalización con  $ENorm=2$



## DATAPROCESADA.INFO

---

El atributo *DataProcesada.Info* corresponde a su vez a un struct con cuatro atributos como se muestra en la siguiente figura.







DataProcesada.Info	
Field ^	Value
 Fuente	"SERIES_Data..."
 ShearSpan	4520
 Unidades	1x2 string
 ID	"Thomsen"

Figura 17: struct *DataProcesada.Info* con sus atributos

### 11.1. DATAPROCESADA.INFO.FUENTE

Corresponde a un string con la fuente de la data experimental cargada

### 11.2. DATAPROCESADA.INFO.SHEARSPAN

Corresponde a un entero con el valor del *ShearSpan* ingresado en la sección de carga de data experimental en el script.

### 11.3. DATAPROCESADA.INFO.UNIDADES

Corresponde a un array con las unidades de la data experimental cargada.

### 11.4. DATAPROCESADA.INFO.ID

Corresponde a un array con el ID del muro asociado a la data experimental cargada.

El atributo *DataProcesada.PerdidaRigidez* corresponde a su vez a un struct con tres atributos como se muestra en la siguiente figura .

DataProcesada.PerdidaRigidez	
Field ^	Value
Carga	4x12 double
Descarga	4x12 double
Secante	2x12 double

Figura 18: struct *DataProcesada.PerdidaRigidez* con sus atributos

## 12.1. NORMALIZACIÓN

La rigidez con la cual se normalizará queda determinada por la variable *KNormOption*.

### ■ *KNormOption*=1

Se utiliza la rigidez del tramo elástico del modelo bilineal ajustado.

### ■ *KNormOption*=2

Esta opción calcula la rigidez secante al punto de la curva esqueleto donde la fuerza es igual a  $Q * F_{max}$ .

Por defecto el valor de  $Q = 0,6$ . A continuación se muestra una figura con el efecto de distintos valores de  $Q$ .

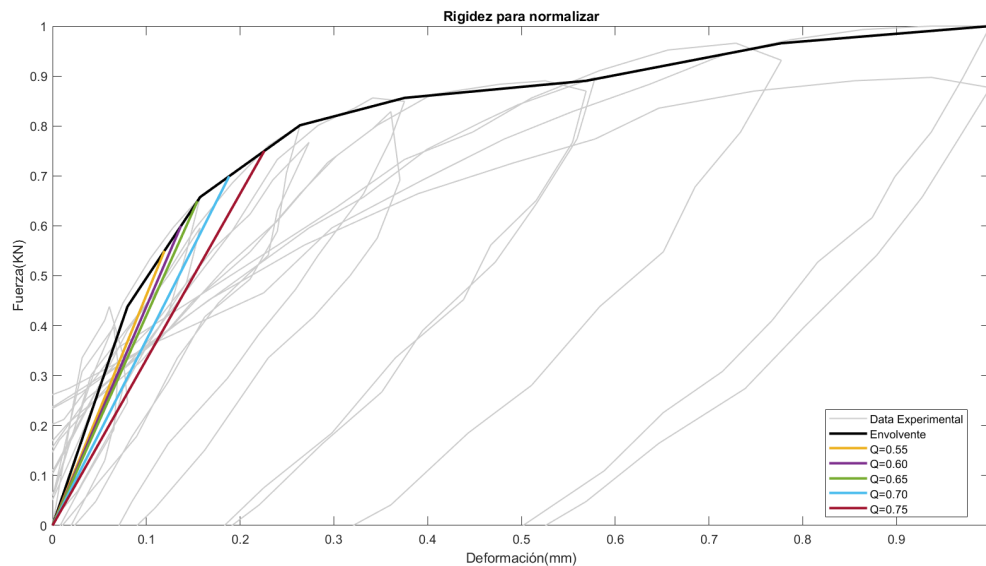


Figura 19: Comparación distintos valores de  $Q$

Similarmente, la rigidez secante considera los siguientes puntos en la curva esqueleto

$$(D, QF_{min}) \quad || \quad (D, QF_{max})$$

Si en cambio, se utiliza el método *KNormOption*=1, la rigidez secante de normalización considera los los puntos:

$$(D_y^-, F_y^-) \quad || \quad (D_y^+, F_y^+)$$

#### 12.2. DATAPROCESADA.PERDIDARIGIDEZ.CARGA

Corresponde a un array donde:

- La primera fila es la rigidez normalizada de carga por el método 1 en la parte positiva en cada ciclos.
- La segunda fila es la rigidez normalizada de carga por el método 2 en la parte positiva en cada ciclo.
- La tercera fila es la rigidez normalizada de carga por el método 1 en la parte negativa en cada ciclo.
- La cuarta fila es la rigidez de carga normalizada por el método 2 en la parte negativa en cada ciclo.

#### 12.3. DATAPROCESADA.PERDIDARIGIDEZ.DESCARGA

Corresponde a un array donde:

- La primera fila es la rigidez normalizada de descarga por el método 1 en la parte positiva en cada ciclos.
- La segunda fila es la rigidez normalizada de descarga por el método 2 en la parte positiva en cada ciclo.
- La tercera fila es la rigidez normalizada de descarga por el método 1 en la parte negativa en cada ciclo.
- La cuarta fila es la rigidez de descarga normalizada por el método 2 en la parte negativa en cada ciclo.

#### 12.4. DATAPROCESADA.PERDIDARIGIDEZ.SECANTE

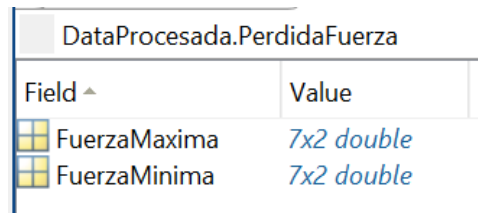
Corresponde a un array donde:

- La primera fila es la rigidez normalizada de secante por el método 1.
- La segunda fila es la rigidez normalizada de secante por el método 2.

## DATAPROCESADA.PERDIDAFUERZA

---

El atributo *DataProcesada.Rigidez* corresponde a su vez a un struct con tres atributos como se muestra en la siguiente figura.



DataProcesada.PerdidaFuerza	
Field ▲	Value
FuerzaMaxima	7x2 double
FuerzaMinima	7x2 double

Figura 20: struct *DataProcesada.PerdidaFuerza* con sus atributos

### 13.1. DATAPROCESADA.PERDIDAFUERZA.FUERZAMAXIMA

Corresponde a un array donde cada columna representa la pérdida de fuerza a ese nivel de deformación (según el protocolo identificado). Cada elemento de un fila corresponde a la razón entre la fuerza máxima en los ciclos a ese nivel de deformación y la fuerza máxima del primer ciclo a ese nivel de deformación.

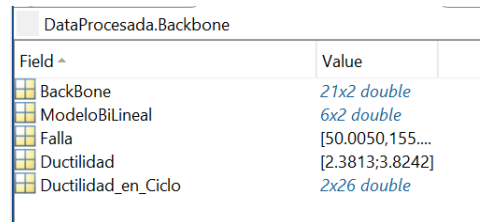
### 13.2. DATAPROCESADA.PERDIDAFUERZA.FUERZAMINIMA

Corresponde a un array donde cada columna representa la pérdida de fuerza a ese nivel de deformación (según el protocolo identificado). Cada elemento de un fila corresponde a la razón entre la fuerza mínima en los ciclos a ese nivel de deformación y la fuerza mínima del primer ciclo a ese nivel de deformación.



## DATAPROCESADA.BACKBONE

El atributo *DataProcesada.Rigidez* corresponde a su vez a un struct con cinco atributos como se muestra en la siguiente figura.



Field ^	Value
BackBone	21x2 double
ModeloBiLineal	6x2 double
Falla	[50.0050,155....
Ductilidad	[2.3813;3.8242]
Ductilidad_en_Ciclo	2x26 double

Figura 21: struct *DataProcesada.Backbone* con sus atributos

### 14.1. DATAPROCESADA.BACKBONE.BACKBONE

El script tiene la capacidad de ajustar 4 tipos de curva esqueleto, el modelo ajustado dependerá del valor de la variable *BackBoneOption*.

A continuación se explican los valores que puede tomar a variables y los modelos asociados a cada valor.

- *BackBoneOption*=1  
Denominado como Deformación máxima\_v1. En este modelo el backbone pasa por el punto de máxima deformación del primer ciclo en cada incremento de deformación.
- *BackBoneOption*=2  
Denominado como el modelo de Fuerza máxima. EL backbone pasa por el punto de fuerza máxima del primer ciclo a cada incremento de deformación.
- *BackBoneOption*=3  
Denominado como el backbone de Deformación máxima \_v2. Se construye considerando que la curva esqueleto pasa por el punto de máxima deformación entre todos los ciclos a un nivel de deformación, para cada incremento de deformación.

### 14.2. DATAPROCESADA.BACKBONE.MODELOBILINEAL

Corresponde a un array con la coordenadas del modelo Bilineal ajustado. El script puede calcular dos tipos de curvas Bi-lineales, lo que depende del valor de a variable *BiLinealOption* En este modelo se iguala la energía que la bilineal sobre-estima en el tramo lineal-elástico con la energía que se subestima en el tramo post fluencia. La tolerancia del error entre estas dos áreas esta definida por la variable. *EnergyTolerance*. Los valores que puede tomar esta variables y los modelos que representa cada valor se explican a continuación

- *BiLinealOption*=1  
Modelo con rigidez post-fluencia, este punto de fluencia es tal que la fuerza en la envolvente a esa deformación sea el  $0,6F_{max}$ .  
A continuación se muestra una figura con este modelo.

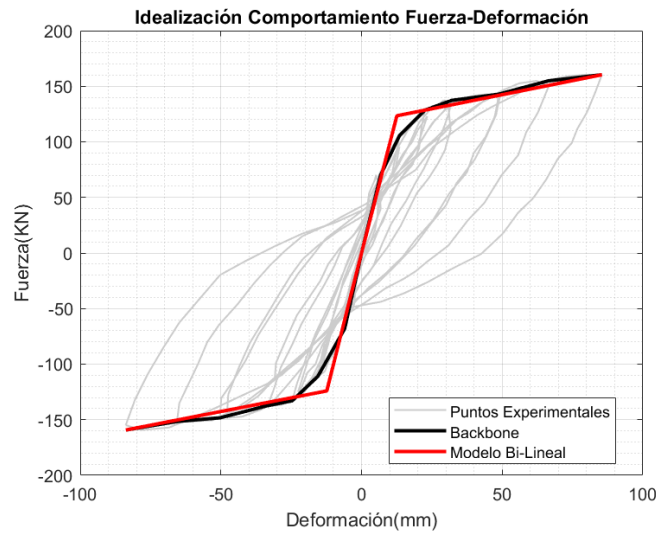


Figura 22: Ajuste modelo bilineal 1

■ *BiLinealOption=2*

Modelo sin rigidez post-fluencia. El punto de fluencia corresponde al punto en la curva esqueleto donde la fuerza es igual al  $0,75F_{max}$ .

A continuación se muestra un ejemplo del ajuste del modelo

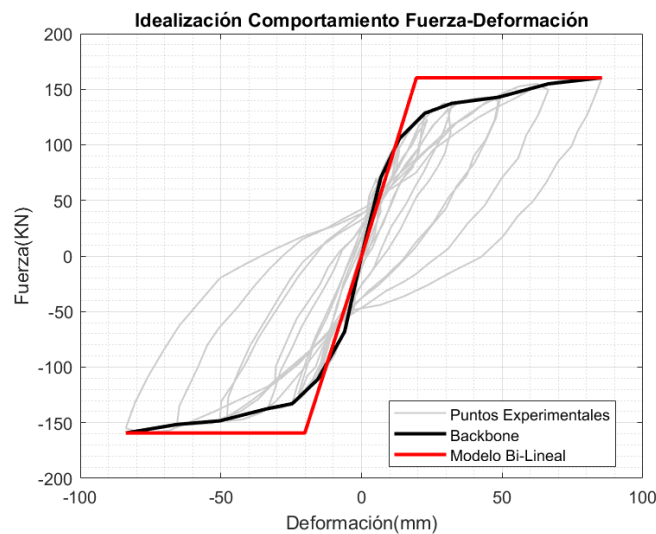


Figura 23: Ajuste modelo bilineal 2

### 14.3. DATAPROCESADA.BACKBONE.FALLA

El Script detecta dos tipos de falla del muro.

■ **Falla por Pérdida de Fuerza a deformación mayor.**

Se define que un muro falla cuando la fuerza a un nivel de deformación es inferior a la fuerza del nivel de deformación anterior. Se establecieron dos posibles casos

- **Ciclos intermedios**

Para niveles de deformación distintos al último que alcanzó al muro. Se establece que el muro falla cuando

$$Fuerza_i < Fuerza_{i-1} * B\_Tolerancia\_intermedia$$

- Ciclo Final

Se establece que el muro falla cuando

$$Fuerza_{end} < Fuerza_{end-1} * B\_Tolerancia\_final$$

Donde  $i$  es el nivel de deformación. Se recomienda establecer la tolerancia intermedia inferior a la tolerancia final para evitar falsos positivos de puntos de falla.

- Falla por Pérdida de Fuerza en mismo nivel de deformación Se establecieron dos posibles casos

- Ciclos intermedios

Para niveles de deformación distintos al último alcanzado por el muro. Se establece que el muro falla cuando

$$\frac{Fuerza_{i,k}}{Fuerza_{i,1}} < B\_tolerancia\_Perdida\_F$$

Donde  $i$  corresponde al nivel  $i$  de deformación y  $k$  al numero del ciclo a ese nivel de deformación

- Ciclo Final

Se establece que el muro falla cuando

$$\frac{Fuerza_{i,k}}{Fuerza_{i,1}} < B\_tolerancia\_Perdida\_F\_final$$

Donde  $i$  corresponde al nivel  $i$  de deformación y  $k$  al numero del ciclo a ese nivel de deformación

Se recomienda establecer la tolerancia intermedia inferior a la tolerancia final para evitar falsos positivos de puntos de falla, especialmente en protocolos con múltiples ciclos a un nivel de deformación.

A continuación se muestra un ejemplo de las detección de puntos de falla.

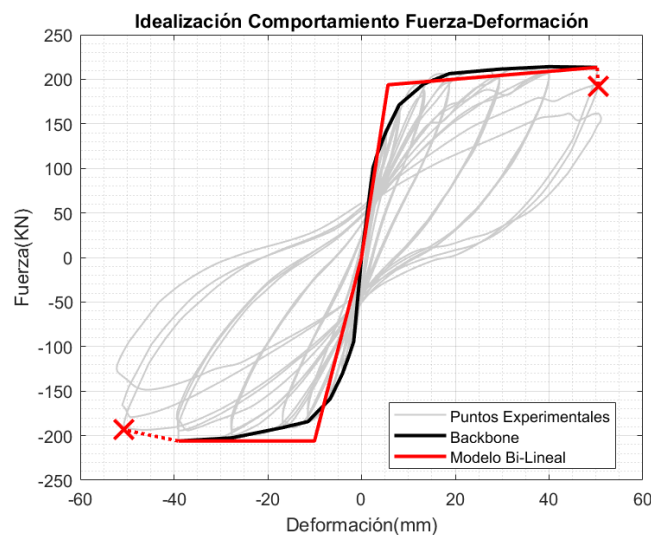


Figura 24: Puntos de falla

En la figura 24 la falla detectada en la parte positiva de la histéresis corresponde a "Falla por Pérdida de Fuerza en Mismo Nivel de Deformación". En cambio, la falla en el lado negativo de la histéresis corresponde a una "Falla por Pérdida de Fuerza a Deformación Mayor". En la figura anterior, el valor de las tolerancias se consideró como 0.98 de manera de poder producir una figura que refleje ambos modos de falla en una sola figura.

#### 14.4. DATAPROCESADA.BACKBONE.DUCTILIDAD

Corresponde a un array de dimensiones 2x1, donde el primer elemento corresponde a la ductilidad de la parte positiva de las histéresis y el segundo elemento a la ductilidad de la parte negativa. La ductilidad se calculó como

$$\mu = \frac{d_u}{d_y}$$

Donde  $d_y$  corresponde a a deformación de fluencia del modelo bi-lineal ajustado y  $d_u$  corresponde a la deformación última. La convención para estimar la deformación última queda determinada por el valor de la variable *DuctilityOption*, específicamente

- *DuctilityOption*=1  
Se considera que la deformación última es el valor máximo del modelo bi-lineal ajustado
- *DuctilityOption*=2  
Si se detectó algún punto de falla, se considera que la deformación última es la deformación asociada al punto de falla. Si no se detectó un punto de falla, se utiliza la convención de *DuctilityOption*=1

## DATAPROCESADA.OTROS

---

El atributo *DataProcesada.Otros* es a su vez un struct con cuatro atributos

### 15.1. *DataProcesada.Otros.Outlier*

Es un logical array con los índices de la data original de los puntos identificados como *outliers*

### 15.2. *DataProcesada.Otros.PuntosNegInicial*

Corresponde un entero que indica la cantidad de puntos que se eliminaron inicialmente pues poseían deformación o fuerza negativa, como parte de lo descrito en la sección pre-procesamiento .

### 15.3. *DataProcesada.Otros.CiclosMuyCortos*

Corresponde a un cell, donde la posición en la primera fila corresponde al numero del ciclo detectado, y categorizado como muy corto, y la segunda fila a un array con los puntos que componen este ciclo.

### 15.4. *DataProcesada.Otros.Comentarios*

Corresponde a un string con el comentario ingresado de en la sección de carga de data experimental.

## ANEXO A. FUNCIONES AUXILIARES

### VALIDACIÓN VISUAL

La validación visual consiste de gráficos que muestran los cambios a la data experimental y los principales resultados. A continuación, se muestran las funciones que realizan esos gráficos sus resultados.

*OutliersGrafico(RawData,Data,outlierindex)*

Esta función gráfica la data original y la data sin *outliers*. Los inputs de esta función son variables intermedias que se utilizan dentro de la rutina principal.

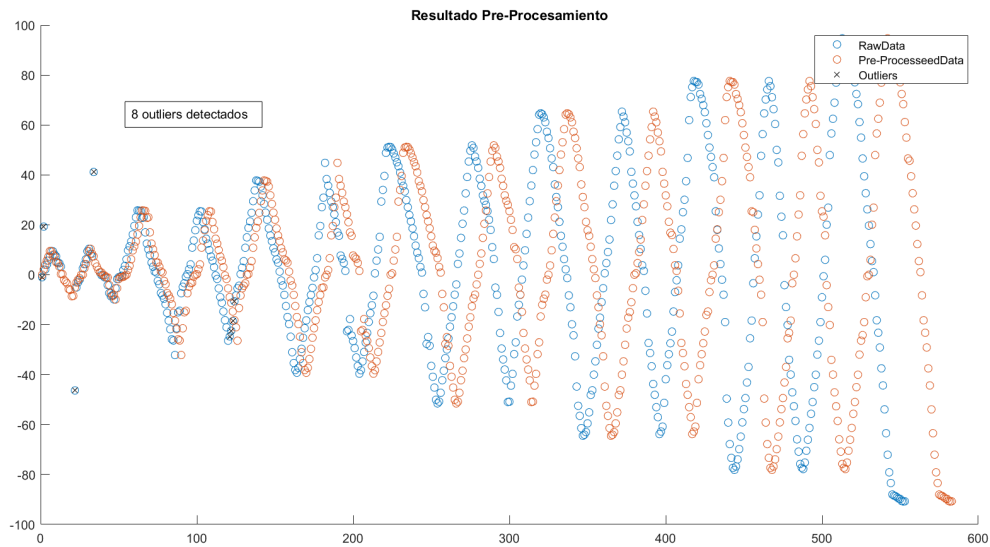


Figura 25: Resultados de detección de *outliers*

*GraficarHisteresis(DataProcesada,n)*

Esta función recibe el *struct* final y un entero, a continuación se presentan los posibles valores del entero y su significado

■  $n = 1$

Se grafica la histéresis con la data procesada, la envolvente el modelo bilineal ajustado.

En la figura 26 se muestra el resultado de la función con  $n = 1$

■  $n = 2$

Se grafican todos los ciclos identificados. Se recomienda tener cuidado con esta opción, pues dependiendo de las características del computador, esto puede demorar múltiples minutos memoria para en ensayos con muchos ciclos ( $k > 50$ ). En la figura 27 se muestra el resultado de la función con  $n = 2$

■  $n = 3$

Se grafican todos los ciclos identificados, las rigideces secantes y todas las rigideces de la parte positiva. Similar a la opción anterior, en ensayos con muchos ciclos, Matlab puede demorarse en crear todos los gráficos. En la figura 28 se muestra el resultado de la función con  $n = 3$

*GraficarCiclos(DataProcesada)*

La función grafica los ciclos detectados y la data experimental en una figura, de manera de poder evaluar

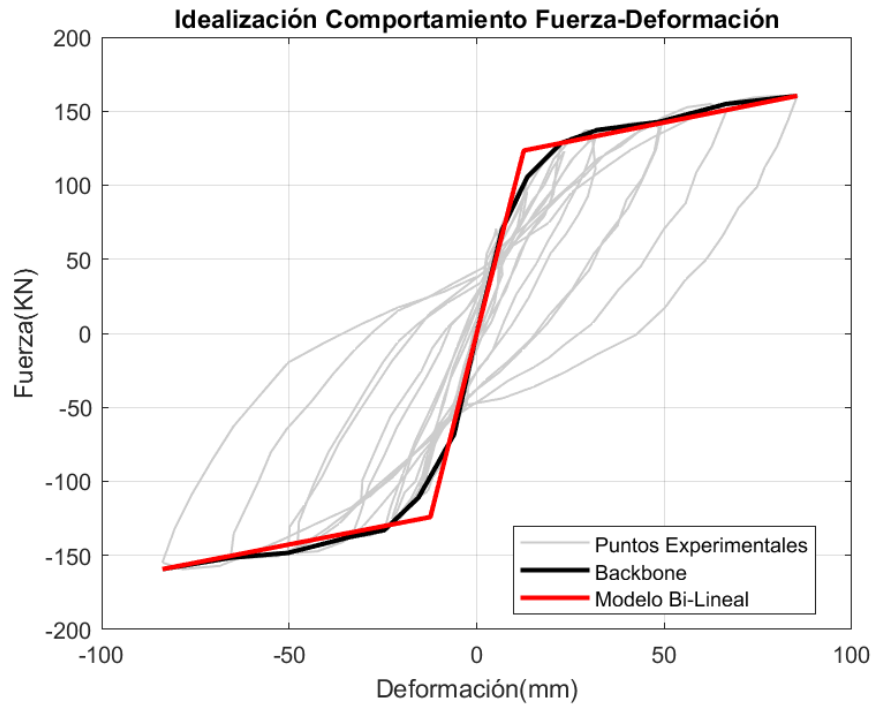


Figura 26: Validación visual de backbone y bilineal

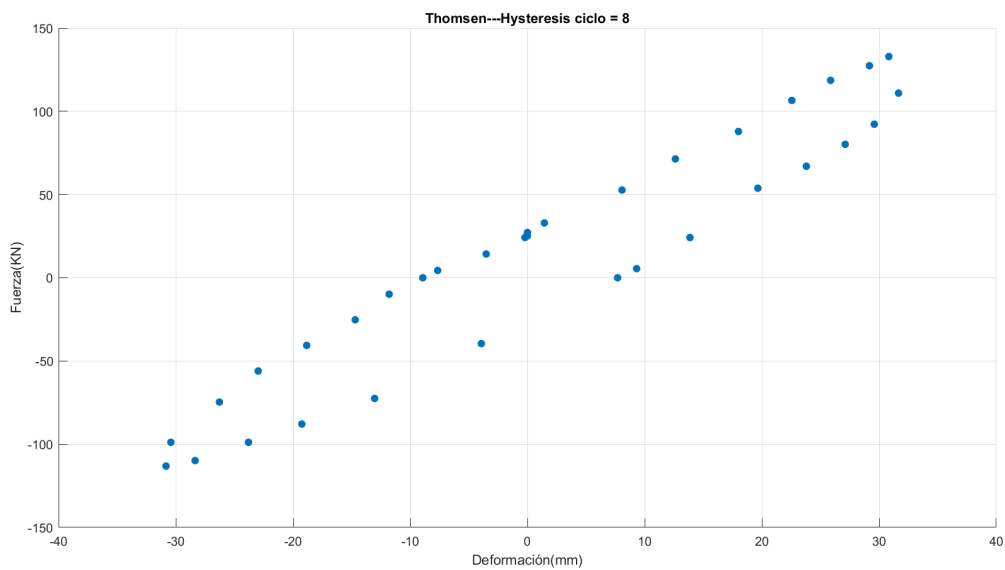


Figura 27: Validación visual de ciclos identificados

rápidamente el procesamiento realizado por el script. En la figura 29 se muestran con distintos colores los ciclos detectados

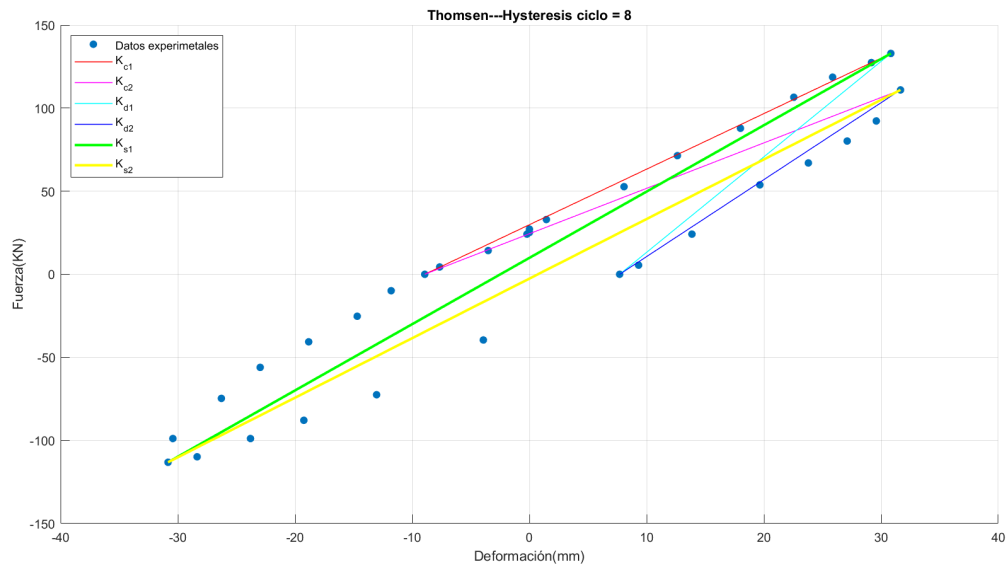


Figura 28: Validación visual de ciclos y rigideces.

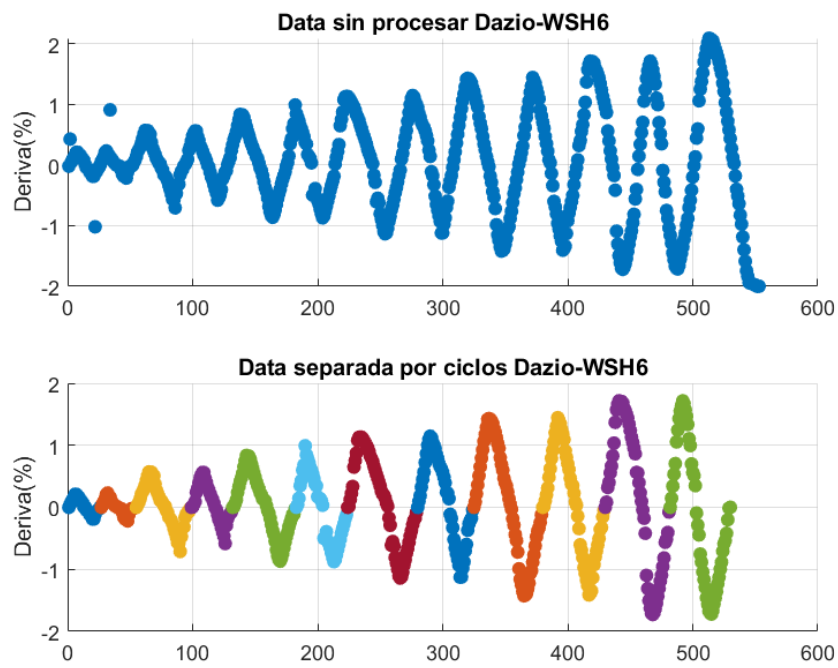


Figura 29: Validación visual de detección de ciclo



### Resultados.m

Esta función gráfica los principales resultados del análisis realizado por el script *Analisis\_Data\_Experimental.m*, en particular, la pérdida de rigidez en función de la deriva y de la energía disipada, como se observa en la figuras 30 y 31.

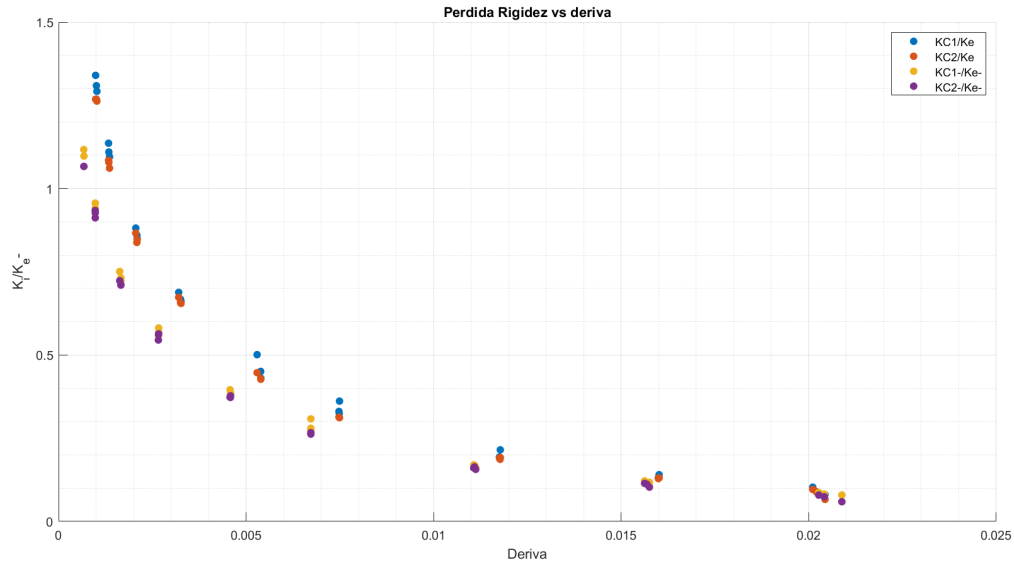


Figura 30: Pérdida de rigidez en función de deriva

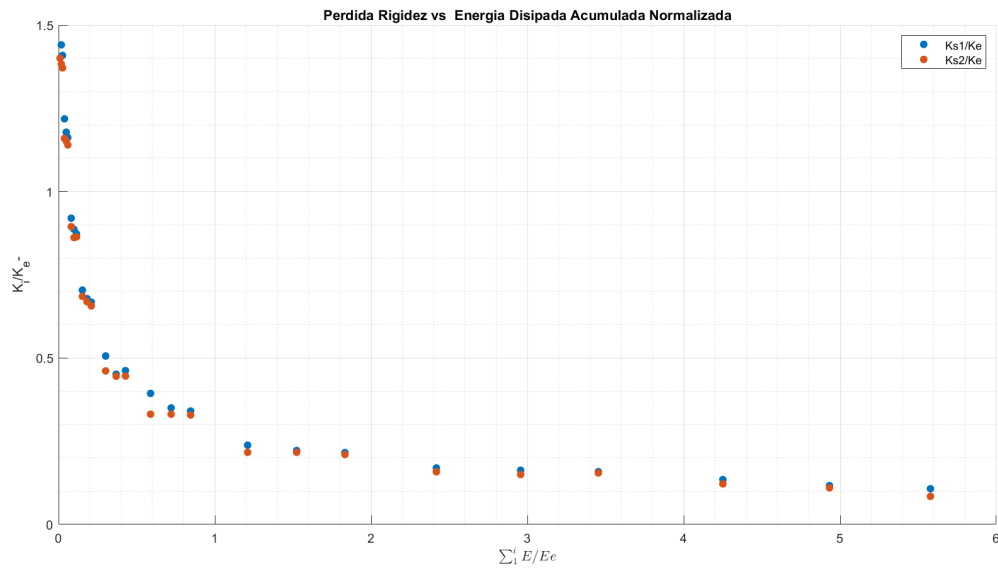


Figura 31: Pérdida de rigidez en función de deriva