



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA CIVIL

COMPARACIÓN DE LA RESPUESTA EXPERIMENTAL Y ANALÍTICA DE MUROS
RECTANGULARES DE HORMIGÓN ARMADO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL

FELIPE IGNACIO CASTRO GONZÁLEZ

PROFESOR GUÍA:
FABIÁN ROJAS BARRALES

MIEMBROS DE LA COMISIÓN:
LEONARDO MASSONE SÁNCHEZ
FRANCISCO HERNÁNDEZ PRADO

SANTIAGO DE CHILE
2017

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL
POR: FELIPE IGNACIO CASTRO GONZÁLEZ
FECHA: 2017
PROF. GUÍA: SR. FABIÁN ROJAS BARRALES

COMPARACIÓN DE LA RESPUESTA EXPERIMENTAL Y ANALÍTICA DE MUROS RECTANGULARES DE HORMIGÓN ARMADO

En el presente trabajo se compara la respuesta experimental de muros rectangulares de diferentes configuraciones de hormigón armado (H.A.), con modelos de elementos finitos generados con elementos cuadriláteros con capa tipo membrana.

Se utiliza la plataforma SAFE-TB, de F. Rojas [11], desarrollada en Matlab, la cual realiza un análisis numérico no lineal de elementos finitos ante cargas verticales estáticas y ante cargas laterales por medio del método pushover cíclico, con el fin de validar la plataforma y verificar las leyes constitutivas de los materiales asumidas.

Cada uno de los muros modelados representan muros experimentales ensayados por otros investigadores, los cuales serán usados como validación de los resultados, replicando sus condiciones de ensayo, cargas dispuestas y ciclos de desplazamiento utilizados.

Para el análisis de los resultados, se comparan las respuestas experimentales con los modelos, utilizando envolventes en ambos casos y comparando la rigidez. Los desplazamientos producto del corte y flexión se estudia de igual forma que en las investigaciones de los muros ensayados.

Se valida la capacidad de la plataforma de modelar muros de H.A., prediciendo correctamente su comportamiento ante desplazamientos impuestos, con distribuciones de esfuerzos y deformaciones, rigidez durante los ciclos por medio de una envolvente, además de lograr analizar los desplazamientos por flexión y corte por separado.

*A mis padres,
por su apoyo y educación*

Agradecimientos

Quisiera agradecer, en primer lugar, a mi familia. Mis padres me han enseñado y cooperado en todo mi proceso formativo y en todos los aspectos de mi vida. A ellos les agradezco este trabajo que finaliza mi educación superior. A mis queridos hermanos, Francisca y Carlos, los cuales han sido un ejemplo a seguir y siempre he contado con ellos para todo. También a mis abuelos, tíos, primos siempre presentes.

A mis amigos del colegio, quienes me han acompañado todos estos años y quienes han logrado ser soporte y parte de todo mi crecimiento a través de los años.

A los de la u, los de la sección y de plan común, especialmente a Santi León F.C., los primeros en ayudarme y estar tanto en el estudio, como en los momentos de relax, además de tantos partidos compartidos. Gracias a cada uno por haberme acompañado y haber hecho los primeros años más fáciles y relajados.

A los civiles, los cuales después de haber escogido la especialidad pasaron a ser mi lugar de estudio, de carrete y de risa, gracias a cada uno por esos incontables trabajos, informes, tareas y hora de estudio que compartimos, además de la compañía y amistad.

A los estructurales, por transformar todo lo cuerdo en una estupidez, sin jamás quitar el mismo interés por la rama que estudiamos, soportando aquellos momentos de estrés pero siempre obteniendo la recompensa de pasarlo bien.

Finalmente, a las demás personas que he conocido en la u, quienes a lo largo de mi vida universitaria me han apoyado y he compartido uno que otro momento, quienes han cambiado mis días, logrando alegrar y mejorarlos, y han logrado hacer este proceso uno inolvidable.

Tabla de Contenido

Índice de Tablas	vi
Índice de Ilustraciones	vii
1. Introducción	1
1.1. Motivación	1
1.2. Objetivo	2
1.3. Alcance	3
1.4. Organización	3
2. Programación y Modelamiento	5
2.1. Programa SAFE-TB	5
2.1.1. Modelo	5
2.1.2. Cargas	6
2.1.3. Análisis	7
2.2. Leyes constitutivas de los materiales	7
2.2.1. Modelo constitutivo del Hormigón	7
2.2.2. Modelo constitutivo del Acero	12
2.3. Cálculo de deformación por flexión y por corte	15
2.3.1. Desplazamiento angular de los elementos	16
2.3.2. Método experimental: Medición de diagonales	16
2.3.3. Método experimental: Desplazamiento vertical en los extremos	17
3. Casos de Estudio	19
3.1. Caso 1 - MR1	19
3.1.1. Investigación: Displacement-Based Design of RC Structural Walls: An Experimental Investigation of Walls with Rectangular And T-Shaped Cross-Sections	19
3.1.2. Espécimen de Estudio: RW2	19
3.1.3. Modelo	23
3.1.4. Análisis y Resultados	25
3.2. Caso 2 - MR2	32
3.2.1. Investigación: Experimental and Analytical Studies of Moderate Aspect Ratio Reinforced Concrete Structural Walls	32
3.2.2. Muro de Estudio: RW-A15-P10-S78	32
3.2.3. Modelo	36
3.2.4. Análisis y Resultados	38

3.3. Caso 3 - MR3	44
3.3.1. Investigación: Quasi-static Cyclic Tests and Plastic Hinge Analysis of RC Structural Walls	44
3.3.2. Espécimen de Estudio: WSH6	45
3.3.3. Modelo	48
3.3.4. Análisis y Resultado	50
3.4. Caso 4 - MR4	55
3.4.1. Investigación: Hysteretic Behavior of Reinforced Concrete Structural Walls	55
3.4.2. Especimen de Estudio: S6	56
3.4.3. Modelo	59
3.4.4. Análisis y Resultados	61
3.5. Caso 5 - MR5	67
3.5.1. Investigación: Earthquake Resistant Structural Walls - Test of Isolated Walls	67
3.5.2. Muro de Estudio: R2	68
3.5.3. Modelo	71
3.5.4. Análisis y Resultados	73
4. Comparación de Modelos	78
5. Recomendaciones de modelo	80
5.1. Mallado	80
5.2. Cargas y Nodo de Control	81
5.3. Procedimiento de control de desplazamiento	81
5.4. Material	81
5.4.1. Hormigón	81
5.4.2. Acero	81
5.5. Coeficiente por confinamiento	82
5.6. Coeficientes de daño	82
5.7. Definición de elementos	82
6. Conclusión	83
Bibliografía	88
A. Figuras	90
B. Códigos de Programación	94

Índice de Tablas

3.1. Geometría Muro RW2	21
3.2. Cuantías de refuerzo en Muro RW2	22
3.3. Características de materiales	22
3.4. Coeficientes de modelo - Modelo MR1	24
3.5. Parámetros de comparación - RW2 vs MR1	29
3.6. Comparación de Rigideces - RW2 vs MR1	29
3.7. Geometría Muro RW-A15-P10-S78	33
3.8. Cuantías de refuerzo en Muro A15-P10-S78	34
3.9. Características de materiales	35
3.10. Coeficientes de modelo - Modelo MR2	37
3.11. Parámetros de comparación - RW-A15-P10-S78 vs MR2	41
3.12. Comparación de Rigideces - RW-A15-P10-S78 vs MR2	41
3.13. Geometría Muro WSH6	45
3.14. Cuantías de refuerzo en Muro WSH6	46
3.15. Características de materiales	47
3.16. Coeficientes de modelo - Modelo MR3	49
3.17. Parámetros de comparación - WSH6 vs MR3	51
3.18. Comparación de Rigideces - WSH6 vs MR3	52
3.19. Geometría Muro S6	57
3.20. Cuantías de refuerzo en Muro S6	58
3.21. Características de materiales	58
3.22. Coeficientes de modelo - Modelo MR4	61
3.23. Parámetros de comparación - S6 vs MR4	63
3.24. Comparación de Rigideces - S6 vs MR4	64
3.25. Geometría Muro R2	69
3.26. Cuantías de refuerzo en Muro R2	70
3.27. Características de materiales	70
3.28. Coeficientes de Daño - Modelo MR5	73
3.29. Parámetros de comparación - R2 vs MR5	75
3.30. Comparación de Rigideces - R2 vs MR5	75

Índice de Ilustraciones

2.1. Modelo del hormigón a compresión	8
2.2. Modelo del hormigón a tracción	9
2.3. Comportamiento cíclico del hormigón	10
2.4. Coeficientes de modificación del hormigón	11
2.5. Relación tensión-deformación del acero de refuerzo	13
2.6. Modelo de comportamiento cíclico de Menegotto-Pinto	14
2.7. Modelo de pandeo de barras, según Massone y Modofér [6] en 2009	15
2.8. Dimensiones utilizadas para el cálculo de deformación por corte	16
2.9. Dimensiones utilizadas para el cálculo de deformación por flexión	18
2.10. Secciones del muro para cálculo de flexión	18
3.1. Montaje experimental ensayado al muro RW2. (Imagen: Wallace, 1995 [16]) .	20
3.2. Muro RW2 en vista 3D. (Imagen: Wallace, 1995 [16])	21
3.3. Disposición de armadura de refuerzo en el muro RW2. (Imagen: Wallace, 1995 [16])	22
3.4. Protocolo de deformación para ensayo del muro RW2. (Imagen: Wallace, 1995 [16])	23
3.5. Mallado Muro MR1	24
3.6. Gráfico de fuerza vs deformación total superior para muro experimental RW2 y modelo MR1	26
3.7. Izq: Envoltente total en muro RW2 y modelo MR1. Der: Promedio de envoltente en muro RW2 y modelo MR1	27
3.8. Izq: Envoltente por flexión en muro RW2 y modelo MR1. Der: Promedio de envoltente por flexión en muro RW2 y modelo MR1	28
3.9. Izq: Envoltente por corte en muro RW2 y modelo MR1. Der: Promedio de envoltente por corte en muro RW2 y modelo MR1	28
3.10. Tensión y deformación del modelo MR1 al 1 % de desplazamiento relativo . .	30
3.11. Tensión y deformación modelo MR1 al d_{max} - 2.21 % de deriva	31
3.12. Montaje experimental ensayado al muro RW-A15-P10-S78. (Imagen: Tran, 2012 [14])	33
3.13. Dimensiones y geometría del muro RW-A15-P10-S78. (Imagen: Tran, 2012 [14])	34
3.14. Sección transversal del muro RW-A15-P10-S78. (Imagen: Tran, 2012 [14]) . .	35
3.15. Protocolo de deformación para ensayo del muro RW-A15-P10-S78. (Imagen: Tran, 2012 [14])	36
3.16. Partición Muro MR2	37
3.17. Ciclo Carga-Desplazamiento MR2 y A15-P10-S78	38

3.18. Izq: Envolverte total en muro RW-A15-P10-S78 y modelo MR2. Der: Promedio de envolverte en muro RW-A15-P10-S78 y modelo MR2	39
3.19. Izq: Envolverte por flexión en muro RW-A15-P10-S78 y modelo MR2. Der: Promedio de por flexión en muro RW-A15-P10-S78 y modelo MR2	40
3.20. Izq: Envolverte por flexión en muro RW-A15-P10-S78 y modelo MR2. Der: Promedio de por flexión en muro RW-A15-P10-S78 y modelo MR2	40
3.21. Tensión y deformación modelo MR2 al 1 % de desplazamiento relativo	42
3.22. Tensión y deformación modelo MR2 al d_{max} - 3.01 % de deriva	43
3.23. Dimensiones del muro WSH6. (Imagen: Dazio, 2009 [3])	46
3.24. Disposición de armadura en el muro WSH6. (Imagen: Dazio, 2009 [3])	47
3.25. Protocolo de deformación para ensayo de muro WSH6. (Imagen: Dazio, 2009 [3])	48
3.26. Partición Muro MR3	49
3.27. Gráfico de fuerza vs deformación total superior para muro experimental WSH6 y modelo MR3	50
3.28. Izq: Envolverte total en muro WSH6 y modelo MR3. Der: Promedio de envolverte en muro WSH6 y modelo MR3	51
3.29. Tensión y deformación modelo MR3 al 1 % de deriva	53
3.30. Tensión y deformación modelo MR3 al d_{max} - 2.22 % de deriva	54
3.31. Ensayo realizado en el muro S6. (Imagen: Vallenas, 1979 [15])	56
3.32. Dimensiones y geometría del Muro S6	57
3.33. Refuerzo de armadura utilizado en Muro S6. (Imagen: Vallenas, 1979 [15])	58
3.34. Protocolo de desplazamiento de ensayo muro S6. (Imagen: Vallenas, 1979 [15])	59
3.35. Partición Muro MR4	60
3.36. Gráfico de fuerza vs deformación total superior para muro experimental S6 y modelo MR4	62
3.37. Izq: Envolverte total en muro S6 y modelo MR4. Der: Promedio de envolverte en muro S6 y modelo MR4	63
3.38. Tensión y deformación modelo MR4 al 1 % de deriva	65
3.39. Tensión y deformación modelo MR4 al d_{max} - 1.64 % de deriva	66
3.40. Ensayo experimental para Muro R2. (Imagen: Oesterle et al., 1980 [8]).	68
3.41. Muro R2 en vista 3D. (Imagen: Oesterle et al., 1980 [8])	69
3.42. Armadura del muro R2. (Imagen: Oesterle et al., 1980 [8])	70
3.43. Protocolo de desplazamiento de ensayo muro R2. (Imagen: Oesterle et al., 1980 [8])	71
3.44. Partición Muro MR5	72
3.45. Gráfico de fuerza vs deformación total superior para muro experimental R2 y modelo MR5	74
3.46. Izq: Envolverte total en muro R2 y modelo MR5. Der: Promedio de envolverte en muro R2 y modelo MR5	75
3.47. Tensión y deformación modelo MR5 al 1 % de deriva	76
3.48. Tensión y deformación modelo MR5 al d_{max} - 3.44 % de deriva	77
A.1. Ciclo Carga-Descarga para muro RW2 y MR2 con modelo de pandeo	90
A.2. Desplazamiento por flexión para modelo MR1, Resultado con nodo 109 y 120 como altura de primer piso.	91
A.3. Desplazamiento por corte para modelo MR1, Resultado con nodo 109 y 120 como altura de primer piso.	91

A.4. Desplazamiento por flexión para modelo MR1, utilizando método por deformación angular del elemento.	92
A.5. Desplazamiento por corte para modelo MR1, utilizando método por deformación angular del elemento.	92
A.6. Desplazamiento por flexión para modelo MR2, utilizando método por deformación angular del elemento.	93
A.7. Desplazamiento por corte para modelo MR2, utilizando método por deformación angular del elemento.	93

Capítulo 1

Introducción

1.1. Motivación

El hormigón armado es el material de construcción más utilizado en Chile. En más del 80 % de las construcciones en altura utilizan muros de hormigón armado (M.H.A.) como su principal elemento resistente. Esto posee su fundamento en el buen comportamiento de este tipo de sistema ante cargas sísmicas, lo cual ha sido observado en los movimientos sísmicos en Chile, como por ejemplo en el terremoto del Maule del 2010 (Mw 8.8). Sin embargo, el hormigón armado posee dos principales problemas: el agrietamiento, que implica perder parte de su capacidad para resistir compresión, y la no-linealidad del acero, afectando el comportamiento a tracción y compresión de este material.

Al suponer el trabajo en conjunto de sus dos principales materiales, hormigón y acero, los muros de H.A. son capaces de resistir los esfuerzos de corte y de flexión, donde el hormigón resiste la compresión y el acero la tracción. El principal problema de esta suposición es que para sismos de gran magnitud los elementos pueden entrar en rangos no lineales, provocando histéresis en el hormigón y el acero, debido a agrietamientos del hormigón y por rangos sobre su fluencia en el caso del acero, además de otros efectos como pandeo, deslizamiento de las barras, entre otros.

Estos dos problemas son de difícil estudio debido a que, normalmente, los muros de hormigón armado forman estructuras enlazadas por elementos (vigas, losas, etc), lo cual no nos permite ver el real comportamiento de los elementos, si no más bien un comportamiento global de las estructuras.

Por este motivo se modela y replica el comportamiento de los muros como elementos aislados, para posteriormente poder incorporar estos modelos a un sistema estructural completo y ver la influencia de estos en los desplazamientos y cargas globales que afectan a la estructura.

Las cargas cíclicas nos permiten ver los ciclos de comportamiento completos de los distintos sectores de los muros, los cuales al sentir compresión y tracción durante el ciclo, reaccionan distinto en un estado de elongación o acortamiento. Son los desplazamientos significativos los

que producen que el acero logre entrar en fluencia, tanto en tracción como en compresión, debilitando y afectando su capacidad de resistir los esfuerzos. A su vez, dado que el hormigón posee una capacidad baja de resistir tracción, sufre fracturas al traccionarse, las cuales aumentan en el caso contrario del ciclo, cuando se ve forzado al aplastamiento debido a la compresión excesiva.

Las cargas cíclicas permiten replicar, en su medida, los efectos de un sismo en los muros, al someterlos a ciclos de cargas y descarga en distintas direcciones.

Bajo este contexto, se modelan muros de hormigón armado y se someten a distintos tipos de cargas, estáticas y cíclicas, deformándose local y globalmente. Para lograr un buen análisis es necesario que los elementos sean robustos, de modo que permitan representar de buena manera la distribución de esfuerzos y deformaciones para distintos tipos de configuraciones.

Con el fin de validar y comparar los modelos, se utilizan investigaciones experimentales en donde se apliquen cargas cíclicas sobre muros rectangulares de hormigón armado, las cuales permiten estudiar los problemas planteados.

Las investigaciones corresponden a ensayos realizados por diferentes investigadores en distintos años, todas disponibles en la literatura; (Tran [14], 2012), (Thomsen and Wallace [16], 1995), (Vallenas et al. [15], 1979), (Oesterle et al. [8], 1980) y (Dazio et al. [3], 2009). Estos ensayos muestran el comportamiento de distintas configuraciones de muros de hormigón armado, ante cargas verticales estáticas y aplicaciones cíclicas de cargas horizontales hasta la falla de los muros.

Los modelos se hacen por medio de la plataforma SAFE-TB desarrollada por ROJAS [11] en el programa computacional MATLAB, la cual permite representar los muros ensayados, modelando los elementos como cuadriláteros con capas de tipo membrana, con curvas de histéresis para cada material que describen la reacción del material para aplicaciones de cargas luego de haber sido cargada en sentido contrario, además de modelar el comportamiento no-lineal de los materiales y incorporar parámetros que cuantifican el daño en el hormigón y acero.

Los casos a estudiar se separan en muros rectangulares con diferentes configuraciones dadas por las dimensiones, cuantías de fierro, tipo de hormigón, cargas aplicadas, particiones de elementos y parámetros de daños.

1.2. Objetivo

Comparar y analizar las respuestas experimental y analítica de diferentes configuraciones de muros rectangulares de hormigón armado.

Objetivos Secundarios

1. Estudiar y comparar la distribución de esfuerzos y deformaciones en los muros estudiados.
2. Validar la eficacia de los elementos cuadriláteros para modelar muros de hormigón armado.
3. Estimar reducciones/aumentos de rigidez para cada caso analizado ante cargas verticales y horizontales.
4. Estudiar el impacto de incorporar el efecto de pandeo de barras en el análisis de los muros estudiados.
5. Analizar la modelación realizada ante distintos casos estudiados.

1.3. Alcance

Esta memoria estudia 5 muros de hormigón armado, los cuales corresponden a muros rectangulares de hormigón armado ensayados previamente. El estudio modela los muros asumiendo consideraciones según el ensayo y el tipo de muro. El análisis se realiza utilizando soluciones de análisis numérico por medio de la modelación de elementos finitos.

1.4. Organización

La presente memoria se distribuye en los siguientes capítulos con el fin de lograr los objetivos presentados:

- Capítulo 2: Programación y Modelamiento

En el presente capítulo se estudian las características principales de la plataforma SAFE-TB. Además, las características de las leyes constitutivas de los materiales utilizados, concreto y acero, los elementos cuadriláteros tipo membrana, utilizados para los modelos aplicados, y los supuestos asumidos para esta investigación.

- Capítulo 3: Casos de Estudio

Para cada uno de los casos de estudio se introduce la investigación, sus características relevantes y propiedades del muro a estudiar. Además, descripción del modelo realizado, los resultados obtenidos por medio del programa y un análisis del caso estudiado.

- Capítulo 4: Comparación de Resultados

Utilizando los resultados de cada modelo, junto a su modelación, se comparan los distintos muros modelados, destacando aquellas usos de coeficientes similares y diferencias según modelo.

- Capítulo 5: Recomendaciones de Modelamiento

Se realiza una pequeña reseña en el trabajo realizado al modelar los diferentes muros, de modo de facilitar el modelamiento de futuras estructuras de H.A. en la plataforma y así facilitar la obtención de resultados y rapidez al trabajar.

- Capítulo 6: Conclusiones

Se verifican los objetivos principales de este trabajo y su cumplimiento. Además, se muestran las conclusiones obtenidas a partir de la comparación entre los resultados experimentales y sus respectivos modelos analizados. Se presentan sugerencias y recomendaciones para posibles mejoras al modelo de cada caso. Adicionalmente, se incluyen posibles errores y aciertos en los modelos realizados.

Capítulo 2

Programación y Modelamiento

Para modelar y representar los muros de hormigón armado es necesario analizar y comprender las consideraciones asumidas en la plataforma, en cuanto a comportamiento de materiales y modelación de los muros. En este capítulo, se revisan las leyes constitutivas utilizadas, las componentes del programa, valores considerados en la modelación y análisis, además de distintos métodos para obtener los desplazamientos debido a corte y flexión separados. Para mayor detalle del programa y la teoría ocupada en el análisis se puede revisar el trabajo de F. Rojas [12] en 2016.

2.1. Programa SAFE-TB

La modelación y análisis de los muros es realizada utilizando el método de elementos finitos, por medio de una plataforma programada en Matlab, SAFE-TB - "Structural Analysis and Finite Elements ToolBox", desarrollada por Rojas [11].

Esta plataforma utiliza la programación orientada a objetos para lograr calcular esfuerzos y desplazamientos en elementos y estructuras. Se utiliza este programa debido a la flexibilidad de la programación, la cual permite variar el modelo y el análisis que se realiza de manera rápida, adaptándose a lo que se estudie y modele, modificando variados aspectos según sea el caso de estudio.

2.1.1. Modelo

Los modelos en este trabajo son realizados por medio de elementos cuadriláteros de tipo membrana. Estos elementos se caracterizan por ser elementos cuadriláteros con 12 grados de libertad (GDL), 2 desplazamientos y 1 giro en el plano por nodo del cuadrilátero. Esto implica que solo se consideran esfuerzos en el plano del elemento ($\sigma_z = \tau_{zx} = \tau_{zy} = 0$).

Los elementos membrana poseen un campo de interpolación para los desplazamientos

sobre él, además de un sistema de capas para el enlace completo entre el acero de refuerzo y el hormigón, con un ángulo de rotación y una formulación definida de la rigidez tangente.

La modelación por capas permite variar e incluir distintos comportamientos en el espesor de un elemento, incorporando materiales y capas de perfiles distintos, logrando replicar las distintas capas que tiene un muro de H.A. (recubrimiento, malla de acero, hormigón confinado), para representar su respuesta ante esfuerzos y deformaciones.

Nodos y elementos

El programa utiliza nodos, que cuentan dentro de sus propiedades, con los grados de libertad y posición. Adicionalmente, en estos se aplican las cargas verticales y horizontales, además, permiten medir e imponer desplazamientos. Estos están unidos entre ellos a través de los elementos.

Los elementos se utilizan para asignar los materiales por sección, definir su comportamiento por zonas, sufrir las fracturas, cambios de rigidez, deformaciones y tensiones.

Las dimensiones de los elementos depende los nodos en sus esquinas. Al crear el mallado para cada ensayo se tuvo en consideración lo siguiente:

- Es recomendable que las dimensiones no bajen de 50 [mm] debido a las condiciones de interpolación del programa.
- Las dimensiones de ancho y altura de elementos se intenta mantener entre 150 [mm] a 200 [mm] para zonas no confinadas del muro, y valores cercanos a 100 [mm] para las zonas confinadas.
- Los elementos deben representar las dimensiones reales del ensayo realizado.

Pedestal de soporte y viga de carga

El soporte inferior de los modelos se compone de elementos con propiedades de hormigón pero completamente elástico, lo cual nos permite asignar las restricciones inferiores de empotramiento y lograr un correcto traspaso de los esfuerzos del muro a las reacciones. En la zona superior de los muros se incorporó una viga de carga con características similares a el soporte inferior, en la cual se aplican cargas horizontales y verticales, y nos permitiría el traspaso de fuerzas al muro. Esto se realiza debido a la configuración de los ensayos, en donde estas dos zonas se mantienen dentro del rango elástico.

2.1.2. Cargas

El modelo utiliza la metodología de control por desplazamiento, es decir, se aplica una deriva lateral en un nodo y se calculan los esfuerzos y deformaciones necesarios para cumplir

el desplazamiento impuesto. Además, cargas estáticas son colocadas inicialmente, según sea el ensayo a representar, para obtener condiciones iniciales antes del desplazamiento lateral.

En los nodos superiores se distribuye la fuerza vertical dependiendo del muro estudiado. Para aquellas investigaciones que no utilizaron carga vertical, se utiliza una carga vertical casi nula, replicando la disposición realizada experimentalmente. En todos los modelos se obtiene el valor de la carga horizontal aplicada superiormente al imponer el desplazamiento en un nodo de control.

Para efectos de este trabajo, se utilizan los desplazamientos máximos y mínimos de los muros experimentales para imponer en los diferentes modelos. En instancias iniciales, se realizan variaciones y más ciclos de carga que los utilizados en los ensayos para ver la respuesta del muro y comprobar el funcionamiento del modelo.

2.1.3. Análisis

La metodología de resolución del análisis es control por desplazamiento usando un proceso iterativo de Newton-Raphson hasta encontrar un punto de convergencia para el desplazamiento seleccionado. Las dimensiones de los elementos finitos a lo largo del muro permite variar la cantidad de ecuaciones a resolver, optimizando los tiempos de solución. A su vez, es posible variar la cantidad de pasos utilizados, variando la precisión y el tiempo de solución.

2.2. Leyes constitutivas de los materiales

En el programa se utilizaron modelos constitutivos de materiales: hormigón y acero de refuerzo. Cada material estaba modelado en forma de capas, las cuales al ser incorporadas a los elementos, conformados por 4 nodos en sus extremos, adquirirían el comportamiento deseado en dependencia a la zona que se buscaba modelar. A continuación, se detallan los dos modelos de materiales utilizados, al igual que las consideraciones específicas asumidas en el presente trabajo.

2.2.1. Modelo constitutivo del Hormigón

El modelo de hormigón armado usado en este trabajo se puede revisar en detalle en el trabajo de F. Rojas [12], en donde se formulan las ecuaciones y teoría utilizada. El presente trabajo utiliza la clase programada *Orthotropic Rotating Angle Concrete Tension Stiffening T 2D Material 01*, la cual representa al modelo de hormigón revisado en esta sección.

El modelo de hormigón corresponde a un material ortotrópico, con relaciones axiales de tensión-deformación en los ejes ortogonales principales, enfocada para lograr representar los agrietamientos en un área. Su comportamiento representa un modelo de hormigón axial en dos direcciones, determinando su respuesta mediante un ángulo de rotación entre los

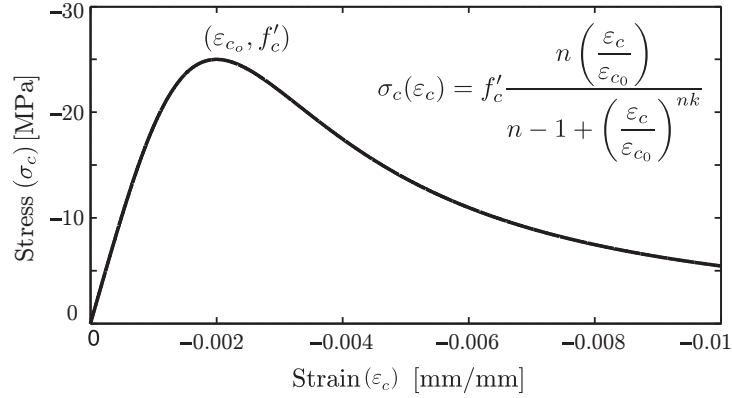


Figura 2.1: Modelo del hormigón a compresión

cuadrantes determinados por los límites $\pi/4$, $-\pi/4$, $3\pi/4$ y $-3\pi/4$. Estas direcciones axiales se determinan a partir de un promedio de las relaciones axiales de tensión-deformación del hormigón.

Las consideraciones básicas asumidas en el modelo corresponden a las siguientes:

- Direcciones principales de tensión y deformación coinciden.
- Relación tensión-deformación puede ser representada por un promedio de tensión-deformación.
- El modelo constitutivo en cada dirección principal es representado por un modelo de hormigón uniaxial.
- El coeficiente de Poisson es nulo posterior al agrietamiento.

A continuación, se muestra el modelo uniaxial formulado por Massone en 2006 [5], extendido para el comportamiento cíclico de histéresis.

Hormigón a compresión

Se define por medio de la curva propuesta por Thorenfeld et al. en 1987 [13], según la ecuación 2.2. En la figura 2.1 se muestra gráficamente el modelo.

$$\sigma_c(\varepsilon_c) = f'_c \frac{n \left(\frac{\varepsilon_c}{\varepsilon_{c0}} \right)}{n - 1 + \left(\frac{\varepsilon_c}{\varepsilon_{c0}} \right)^{nk}} \quad (2.1)$$

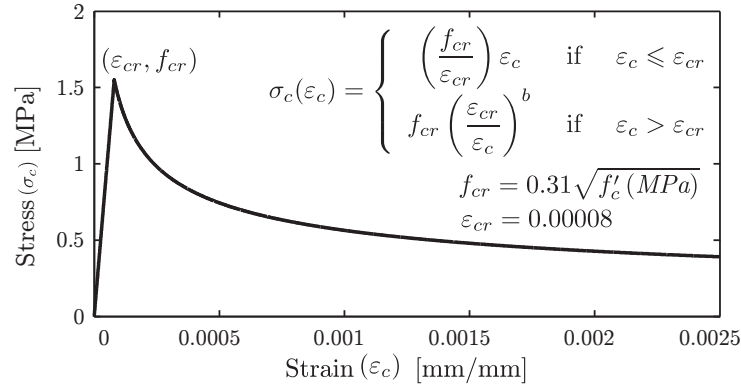


Figura 2.2: Modelo del hormigón a tracción

Para todos los modelos se consideraron valores de $\nu = 0.2$ y $\gamma_h = 2500[kg/m^3]$. La deformación unitaria en la fibra más comprimida del hormigón ε_c , se considera igual $\varepsilon_c = -0.003$ en el centro no confinado y en la zona confinada $\varepsilon_c = -0.0055$.

En el modelo a compresión del hormigón se utilizaron valores de $k = 1$ y $n = 3.1$ para los elementos confinados, $k = 1$ y $n = 2.1$ para aquellos no confinados, propuestos por Carreira y Kuang-Han [2] en 1985 para un valor de f'_c igual a 26.5 [MPa] y 37.5 [MPa] en la siguiente ecuación:

$$n = 1.55 + \left(\frac{f'_c(MPa)}{32.4} \right)^3 \quad (2.2)$$

Hormigón a tracción

Se utiliza el modelo propuesto por Berlabi y Hsu [1] en 1994, descrito por las ecuación 2.3. En la figura 2.2 se muestran las ecuaciones para determinar los valores de f_{cr} y ε_{cr} .

$$\sigma_c(\varepsilon_c) = \begin{cases} \left(\frac{f_{cr}}{\varepsilon_{cr}} \right) \varepsilon_c & \text{si } \varepsilon_c \leq \varepsilon_{cr} \\ f_{cr} \left(\frac{\varepsilon_{cr}}{\varepsilon_c} \right)^b & \text{si } \varepsilon_c > \varepsilon_{cr} \end{cases} \quad (2.3)$$

Los valores utilizados fueron de $b = 0.4$ para los elementos confinados y $b = 0.6$ para los elementos no confinados, recomendados por Berlabi y Hsu [1]. Los valores de ε_{cr} y f_{cr} fueron determinados por la ecuación de Berlabi y Hsu [1] señaladas a continuación:

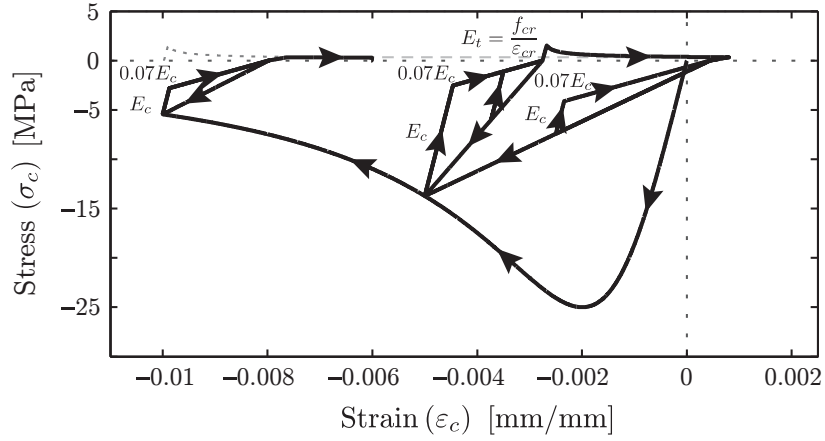


Figura 2.3: Comportamiento cíclico del hormigón

$$f_{cr} = 0.31\sqrt{f'_c} [MPa] \quad (2.4)$$

$$\varepsilon_{cr} = 0.00008 \quad (2.5)$$

Histéresis

Para el comportamiento cíclico del hormigón, se utilizaron las ecuaciones 2.6 y 2.7, para la deformación plástica en cada descarga a compresión y a tracción respectivamente. En la figura 2.3 se puede ver una representación de descarga y recarga a compresión, en donde las pendientes de recarga vienen dadas por E_{c0} y $0.07E_{c0}$ desde el punto de descarga, valores recomendados por Palermo y Vecchio [10] en 2003.

$$\varepsilon_p^c = \varepsilon_m^c \left(1 - e^{-\alpha_c \left(\frac{\varepsilon_m^c}{\varepsilon_{c0}^c} \right)} \right) \quad (2.6)$$

$$\varepsilon_p^c = \varepsilon_m^c \left(1 - e^{-\alpha_c \left(\frac{\varepsilon_m^c}{\varepsilon_{c0}^c} \right)} \right) \quad (2.7)$$

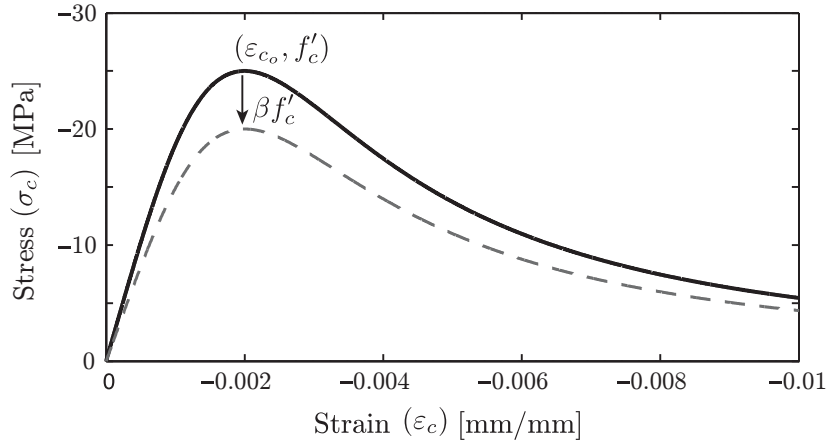


Figura 2.4: Coeficientes de modificación del hormigón

En el trabajo de Rojas [12] se consideraron valores de $\alpha_c = 0.32$ y $\alpha_t = 0.08$, los cuales se ocuparon en el presente trabajo.

Coeficientes de modificación al modelo de hormigón

En el modelo se consideran distintos coeficientes β de aumento y reducción de la capacidad del hormigón. Todos modifican el valor de f'_c como se muestra en la figura 2.4.

a) Coeficiente de reducción por daño

El modelo considera un coeficiente de reducción de la capacidad del hormigón cuando éste sufre compresión biaxial y daño para cargas cíclicas.

El efecto del daño debido a cargas cíclicas o reversibles del hormigón $\beta_d f'_c$, revisado en detalle en el capítulo 3.1.2. de Rojas [12], nos permite cuantificar el deterioro del concreto ante estos ciclos.

La ecuación utilizada fue propuesta por Palermo y Vecchio [10] como:

$$\beta_d = \frac{1}{1 + \alpha_1 \left(\frac{\varepsilon_{rec}}{\varepsilon_{c0}} \right)^{\alpha_2}} \quad (2.8)$$

Donde ε_{c0} corresponde a la tensión de resistencia máxima y ε_{rec} a la diferencia entre la tensión máxima y mínima durante los ciclos. Para α_1 se recomienda valores de 0.1 para antes

del fisuramiento y 0.175 posterior a fisuramiento, y para α_2 se recomienda 0.5 para antes del fisuramiento y 0.6 posterior al fisuramiento, aunque durante el análisis de los muros pasa a ser un parámetro a ajustar.

b) Coeficiente de aumento por confinamiento del hormigón

A partir del confinamiento, se considera un factor de β_{conf} , aplicado en el esfuerzo máximo a compresión f'_c . El valor fue probado para $\beta_{conf} = 1.1$, y para aquel calculado con la ecuación de Orakcal et al. [9] en 2006, señalada en 2.9.

$$\beta_{conf} = 1 + \frac{\phi_{hoops} F_{y_{hoops}}}{f'_c} \quad (2.9)$$

c) Coeficiente de ablandamiento del hormigón

Este coeficiente se debe al ablandamiento en la resistencia del hormigón producto de una tensión en la dirección perpendicular. Se utiliza la ecuación formulada por Berlabi y Hsu [1], en donde se recomiendan valores de $k = 0.9$ y $k_\theta = 250$, aunque en el trabajo de Rojas [12] se utilizan valores de $k = 1$ y $k_\theta = 200$ debido a la estabilidad sin significativa pérdida de precisión.

$$\beta_s = \frac{k}{\sqrt{1 + k_\theta \varepsilon_{tensin}}} \quad (2.10)$$

2.2.2. Modelo constitutivo del Acero

El modelo de acero de refuerzo embebido en hormigón utilizado en este trabajo, se puede revisar en detalle en trabajo de F. Rojas [12]. El presente trabajo se utiliza la clase programada *Smeared Embedded Steel Double Layer MP T 2D Material 01*, la cual representa al modelo de acero revisado en esta sección.

El acero es modelado como capas homogéneas dentro del hormigón, en donde las variaciones de su tensión fueron modeladas por el promedio de la relación tensión-deformación en el área del elemento. Además, las barras de acero solo actúan en su orientación principal.

Las consideraciones asumidas en el modelo corresponden a las siguientes:

- Acero se considera embebido y solo actúa en su dirección axial.
- Relación tensión-deformación puede ser representada por un promedio de tensión-deformación.
- Hormigón y acero se consideran completamente unidos.
- El coeficiente de Poisson es nulo posterior al agrietamiento.

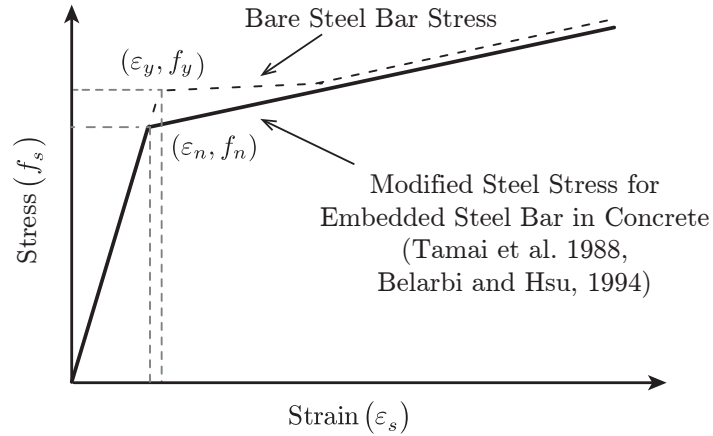


Figura 2.5: Relación tensión-deformación del acero de refuerzo

El modelo utilizado corresponde al modelo de Menegotto y Pinto [7] formulado en 1973, incluyendo un modelo de Filippou [4] de endurecimiento isotrópico. Este incluye la formulación tangencial usada en elementos finitos para análisis estructural.

Modelo uniaxial del acero de refuerzo

Se observa que el acero tiene diferencias de comportamientos cuando este está embebido en hormigón y cuando está sin presencia de hormigón. El efecto se representa como una reducción del esfuerzo de fluencia f_y , seguido por un aumento por endurecimiento de las barras al estar embebidas de hormigón, como se muestra en la figura 2.5.

El valor de E_s y F_y para cada barra de acero fue correspondiente al valor de ensayo en cada investigación. Para aquellas que no tenían el valor del módulo de Young, se utilizó $E_s = 200,000 [MPa]$.

El modelo de acero uniaxial escogido corresponde al modelo de Menegotto-Pinto de 1973 [7], modificado por Filippou et al. en 1963 [4], modelo utilizado en el programa por su eficiencia computacional y su capacidad de representar el comportamiento de las barras de acero. La representación gráfica del modelo está en la figura 2.6.

La curva de interacción está definida por la ecuación 2.11, donde b representa un coeficiente de endurecimiento por deformación, y R y ξ están definidos por las ecuaciones 2.12 y 2.13. Los valores utilizados fueron de $R_0 = 18$, $a_1 = 16.2$ y $a_2 = 0.15$, recomendados en el trabajo de Rojas [12].

$$f_s^*(\varepsilon_s^*) = b\varepsilon_s^* + \frac{(1-b)\varepsilon_s^*}{(1+(\varepsilon_s^*)^R)^{1/R}} \quad (2.11)$$

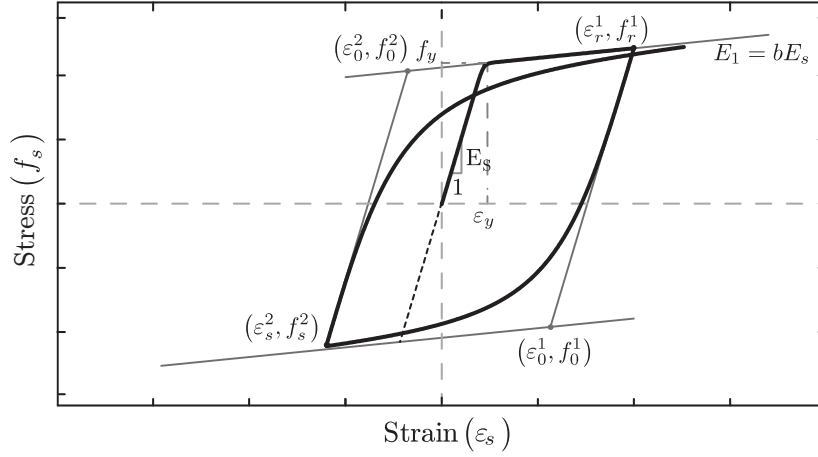


Figura 2.6: Modelo de comportamiento cíclico de Menegotto-Pinto

$$\xi = \left| \frac{\varepsilon_m - \varepsilon_0}{\varepsilon_y} \right| \quad (2.12)$$

$$R = R_0 - \frac{a_1 \xi}{a_2 + \xi} \quad (2.13)$$

Las rectas asintotas quedan definidas con pendiente E_s a partir de la última posición de la carga de retorno ε_r y f_r . El punto de interceptación de ambas rectas es el punto f_0 y ε_0 , como se muestra en la figura 2.11.

$$\varepsilon_s^* = \frac{\varepsilon_s - \varepsilon_r}{\varepsilon_0 - \varepsilon_r} \quad (2.14)$$

$$f_s^* = \frac{f_s - f_r}{f_0 - f_r} \quad (2.15)$$

Para el promedio de tensión-deformación al usar el modelo uniaxial se necesitan modificar los valores por f_y y b por un promedio, como lo señala Berlabi y Hsu [?] en el 1994, con las siguientes relaciones:

$$f_y^{Promedio} = (0.91 - 2B)f_y \quad (2.16)$$

$$b^{Promedio} = (0.02 + 0.25B)b \quad (2.17)$$

Donde B queda definido por $B = \frac{(f_{cr}/f_y)^{1.5}}{\rho}$.

Modelo uniaxial del acero Massone Moroder con pandeo

En ciertos casos de estudio se utilizó un modelo con efectos de pandeos para ver el efecto de estos modelos en los resultados, comparados con el modelo anteriormente descrito. Este modelo aparece descrito en detalle en el trabajo de [12], y esta definido por el efecto descrito en la figura 2.7. El modelo de acero fue utilizado en el análisis experimental realizado por Yañez [17] en 1993.

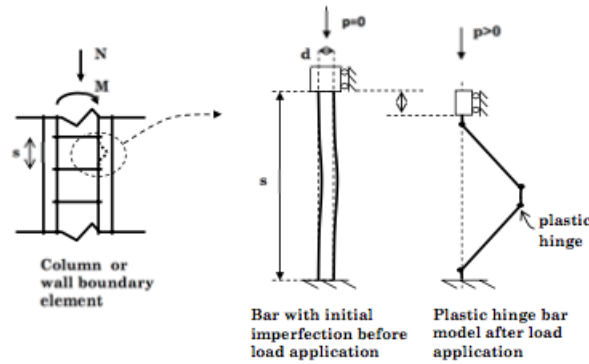


Figura 2.7: Modelo de pandeo de barras, según Massone y Modofer [6] en 2009

2.3. Cálculo de deformación por flexión y por corte

Se utilizaron tres diferentes metodologías para calcular la deformación por corte y por flexión, siempre considerando la validez de la ecuación 2.18.

$$\Delta_{total} = \Delta_{flexion} + \Delta_{corte} + \Delta_{Base} \quad (2.18)$$

La primera forma de obtener estos desplazamientos se obtiene de la deformación angular de los elementos, la segunda del cálculo utilizado por Thomsen, J. y Wallace J. (1995) [16] en su investigación, y la última utilizada por Thien Anh Tran (2012) [14]. En cada caso, se calculó utilizando las tres metodologías, pero se presenta aquella que corresponda en cada caso.

A continuación se muestra el procedimiento de cálculo de las tres opciones comentadas anteriormente.

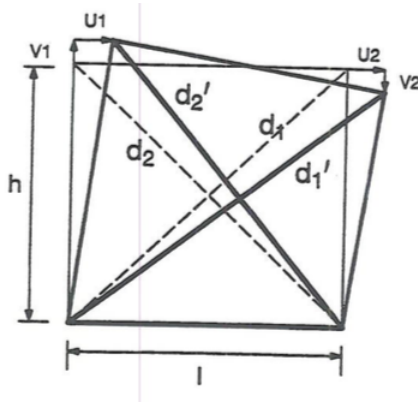


Figura 2.8: Dimensiones utilizadas para el cálculo de deformación por corte

2.3.1. Desplazamiento angular de los elementos

Para el cálculo de la deformación producida solo por el corte en los elementos o muro, se realizaron los siguientes pasos:

1. En cada elemento, se promedió la deformación angular de sus nueve puntos de cuadratura.
2. Se multiplicó por la altura de cada elemento para obtener el desplazamiento en cada elemento producto del corte
3. Este desplazamiento se promedió entre todos los elementos ubicados a la misma altura.
4. Finalmente, se sumaron estos desplazamientos entre todas las alturas y se obtuvo el desplazamiento en el punto medido producto del corte.

Para obtener la deformación por flexión se realiza la resta entre el desplazamiento total y el desplazamiento por corte.

2.3.2. Método experimental: Medición de diagonales

Este método fue utilizado en la investigación de Thomsen, J. y Wallace J. (1995) [16], el cual consistía en determinar el desplazamiento angular por medio de los desplazamientos relativos entre los nodos dispuestos en el muro.

Según la figura 2.8, podemos determinar las distancias d_1' y d_2' usando las siguientes ecuaciones:

$$d_1' = \sqrt{(h - V_2)^2 + (l + U_2)^2} \quad (2.19)$$

$$d_2' = \sqrt{(h + V_1)^2 + (l - U_1)^2} \quad (2.20)$$

Y con ellas determinar el desplazamiento angular γ_{avg} por medio de:

$$2 \cdot \gamma_{avg} = \frac{(d'_1 - d_1)d_1}{lh} + \frac{(d_2 - d'_2)d_2}{lh} \quad (2.21)$$

$$\Delta_{corte} = \gamma h \quad (2.22)$$

$$\Delta_{flexin} = \Delta_{total} - \Delta_{corte} \quad (2.23)$$

2.3.3. Método experimental: Desplazamiento vertical en los extremos

El calculo realizado por Thien Anh Tran (2012) [14] determinó el desplazamiento por flexión utilizando la siguiente ecuación:

$$\Delta_{flex} = \alpha \theta h \quad (2.24)$$

utilizando un valor de θ , calculado experimentalmente, por medio del desplazamiento vertical de los nodos a los extremos para una determinada altura h , como se muestra en la figura 2.9.

$$\theta = \frac{V_1 - V_2}{L} \quad (2.25)$$

Los valores del coeficiente α corresponden a la ubicación del centroide de la distribución de curvatura a lo largo de la sección de muro. Este valor fue determinado experimentalmente en la investigación de Tran [14] en 2012, dividiendo el muro en 4 secciones de A a D como se muestra en la figura 2.10, con valores de α igual a 8/15, 5/9 y 2/3

Finalmente, el valor del desplazamiento por flexión es la suma de los desplazamientos de cada piso.

$$\Delta_{corte} = \Delta_{total} - \Delta_{flexion} \quad (2.26)$$

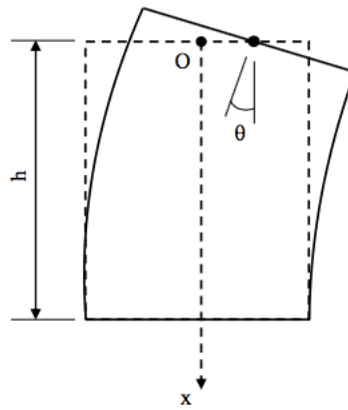


Figura 2.9: Dimensiones utilizadas para el calculo de deformación por flexión

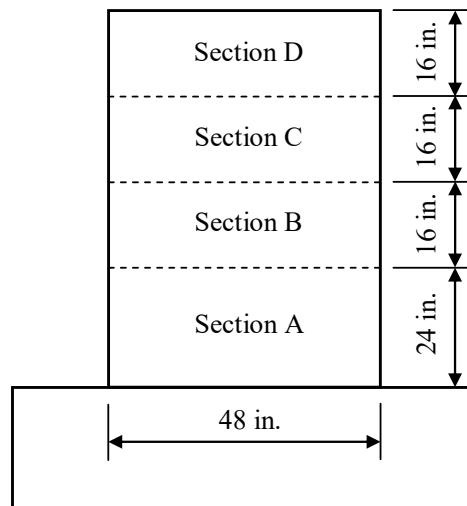


Figura 2.10: Secciones del muro para cálculo de flexión

Capítulo 3

Casos de Estudio

3.1. Caso 1 - MR1

3.1.1. Investigación: Displacement-Based Design of RC Structural Walls: An Experimental Investigation of Walls with Rectangular And T-Shaped Cross-Sections

Thomsen, J. y Wallace J. (1995) [16] realizaron una investigación experimental y analítica sobre cuatro muros de hormigón armado sujetos a cargas axiales verticales y laterales cíclicas.

Los objetivos de la investigación se centraron en verificar un diseño basado en desplazamiento para muros de hormigón armado y ver el comportamiento de muros estructurales asimétricos.

Los muros investigados fueron cuatro, dos de sección rectangular y dos de sección T, todos diseñados a 1/4 de escala. Las diferencias entre los ensayo fueron la forma transversal, su configuración y disposición del refuerzo transversal, y la distribución del refuerzo vertical y horizontal del alma del muro.

Se utiliza el muro RW2 de esta investigación para ser analizado en el presente trabajo. Al modelo de este muro se le asigna el nombre MR1, del cual se obtendrá su respuesta analítica ante similares desplazamientos que el muro RW2, y se compararán con los resultados experimentales.

3.1.2. Espécimen de Estudio: RW2

El muro RW2 es un muro de hormigón armado rectangular, constituido por 4 pisos de igual altura y apoyado sobre un pedestal. El muro posee una zona superior de igual ancho pero mayor armadura que el muro, donde es cargado verticalmente de forma estática, y

horizontalmente por un costado de forma cíclica. El montaje experimental se puede observar en la figura 3.1. El muro tuvo comportamiento a flexión, y se detuvo el ensayo cuando se produjo un exceso de aplastamiento de las zonas confinadas. El ensayo experimental del muro tuvo primeros agrietamientos por flexión ante una deriva superior de 0.1 % a 0.25 %, logrando la fluencia para una deriva de 0.58 %. El muro alcanzó su máxima resistencia para una deriva de 2.35 %, deteniendo el ensayo al notar un grave aplastamiento en las zonas confinadas inferiores producto del aplastamiento, denotando un comportamiento por flexión a lo largo del ensayo.

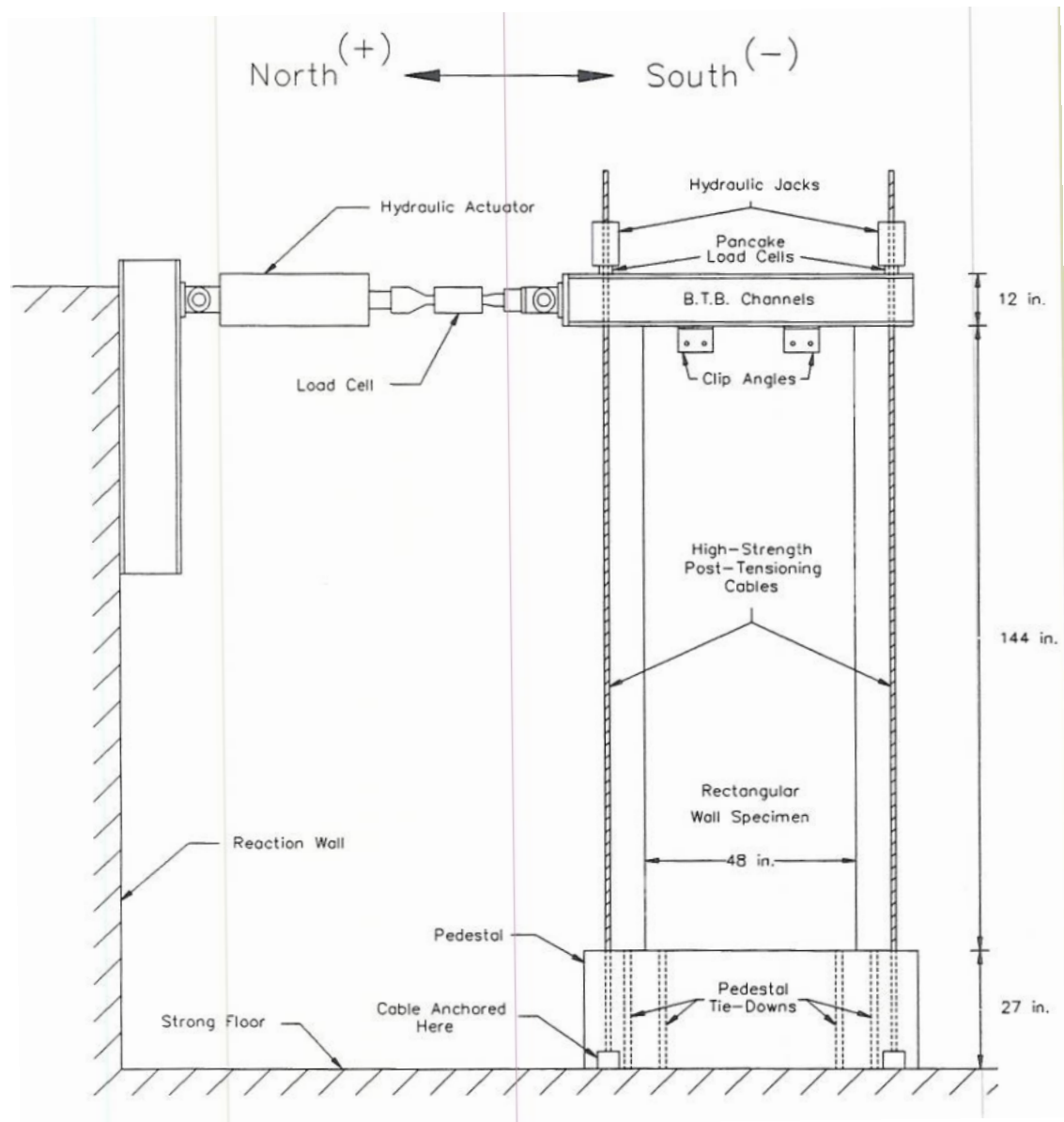


Figura 3.1: Montaje experimental ensayado al muro RW2. (Imagen: Wallace, 1995 [16])

Dimensiones y Geometría

La altura, ancho y espesor del muro se especifican en la tabla 3.1. En la figura 3.2 se puede ver el muro de forma gráfica.

Tabla 3.1: Geometría Muro RW2

Dimensión	Valor
$H_w [mm]$	3660
$L_w [mm]$	1220
$L_{conf} [mm]$	191
$T_w [mm]$	102
H_w/L_w	3.00
L_w/T_w	11.96
s/d	5.33

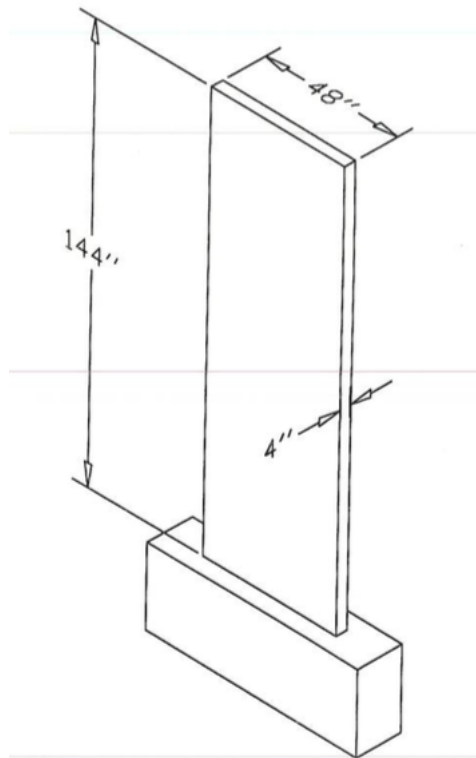


Figura 3.2: Muro RW2 en vista 3D. (Imagen: Wallace, 1995 [16])

Armadura de refuerzo

En la tabla 3.2 se muestran las cuantías de refuerzo, en porcentaje, según sección de muro. Las disposiciones de la armadura en su sección transversal se muestra en la figura 3.3.

Tabla 3.2: Cuantías de refuerzo en Muro RW2

Cuantía	[%]
ρ_v	0.30
ρ_h	0.33
$\rho_{estribos}$	0.69
ρ_{conf}	2.93

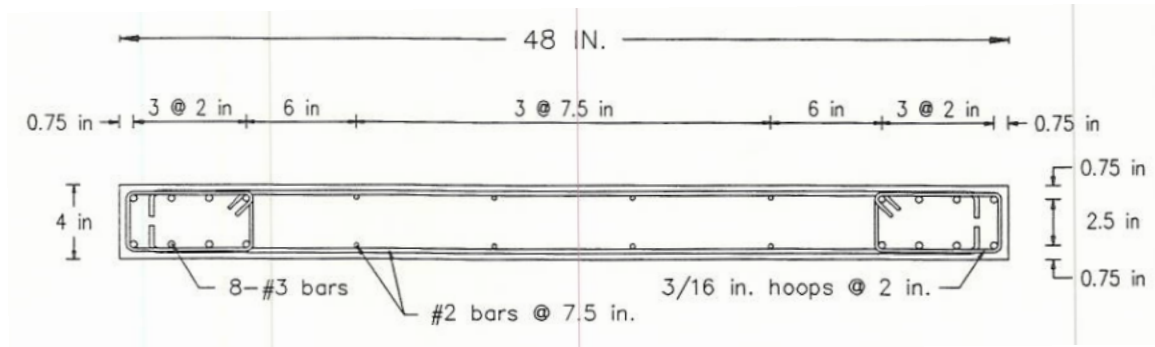


Figura 3.3: Disposición de armadura de refuerzo en el muro RW2. (Imagen: Wallace, 1995 [16])

Materiales

Para cada uno de los materiales, tanto barras de acero como hormigón, se especifican las capacidades de resistencia y módulo de Young utilizados para el diseño y obtenidos experimentalmente para el muro RW2, presentados en la tabla 3.3.

Tabla 3.3: Características de materiales

Característica	Diseño	Experimental
f'_c [MPa]	27.6	43
f_{yv} [MPa]	414	448
f_{yh} [MPa]	414	448
f_{yconf} [MPa]	414	434
f_{ys} [MPa]	414	434
E_s [MPa]	200,000	[-]*

*: La investigación o reporte de los ensayos no presentó datos del módulo de Young calculado experimentalmente.

Cargas aplicadas

La carga vertical estática correspondía a $-378,099[N]$, o $P/(A_g \cdot f'_c) = 0.07$, distribuidos a lo largo de la zona superior. La carga horizontal se aplicó utilizando un protocolo de carga

cíclico, realizando dos ciclos al mismo nivel de deformación antes de aumentar la deformación lateral. Posterior al agrietamiento, el ensayo fue controlado por la deriva superior Δ . En la figura 3.4 se muestra el protocolo de deformación utilizado en el ensayo.

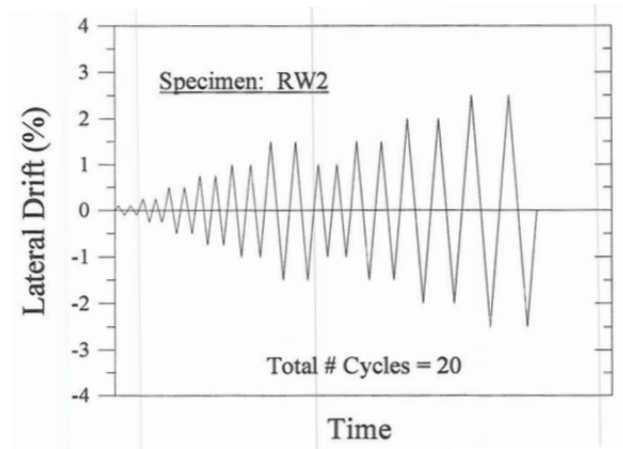


Figura 3.4: Protocolo de deformación para ensayo del muro RW2. (Imagen: Wallace, 1995 [16])

3.1.3. Modelo

Mallado

El muro se modeló con diez elementos horizontales, dos elementos a cada extremo representa la zona confinada, con anchos de 95 [mm] y 96 [mm], y otros seis elementos centrales de ancho 140 [mm]. La partición vertical consta de 17 elementos de altura 215 [mm], seguidos por otros 4 elementos de altura 76 [mm]. El pedestal de fundación está constituido por cuatro elementos verticales de altura 215 [mm] y 14 elementos horizontales de ancho similar al dispuesto en los elementos de muro adyacentes. En la figura 3.5 se puede ver el mallado de forma gráfica.

Cargas

Las fuerzas verticales actuantes sobre el modelo se ejercieron en los nueve nodos centrales superiores con una carga de -378 [kN] distribuida en los nodos centrales superiores. Las cargas horizontales fueron aplicadas en los tres nodos izquierdos superiores, señalados en rojo en la figura 3.5. En verde, el nodo de registro de desplazamientos.

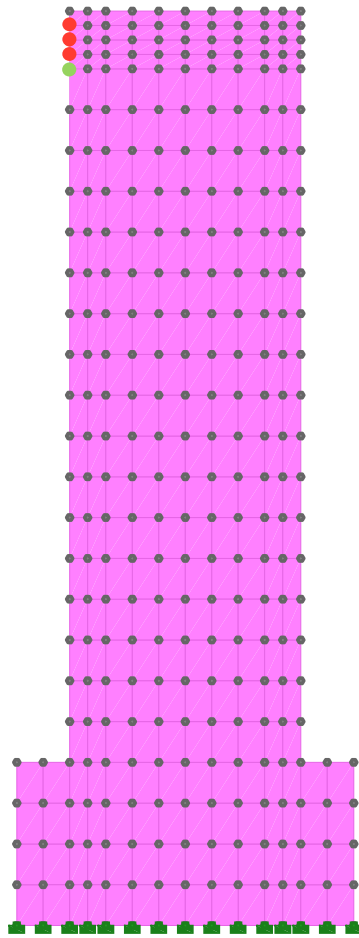


Figura 3.5: Mallado Muro MR1

Coefficientes de daño

Los coeficientes de daño utilizados en la zona confinada y no confinada se señalan en la tabla 3.4.

Tabla 3.4: Coeficientes de modelo - Modelo MR1

Coeficiente	Valor
α_{1conf}	0.1
α_{2conf}	0.5
α_1	0.1
α_2	0.5
K_{conf}	1.06

3.1.4. Análisis y Resultados

El comportamiento del muro experimental RW2 fue dominado por flexión, debido a su relación de aspecto de 3.0. Además, presentaba una relación $s/d=5.33$, la cual no presentó pandeo de sus barras por mal confinamiento, si no hasta los ciclos finales de desplazamiento por aplastamiento excesivo de la zona. El ensayo culminó antes de la falla total del muro, es decir, antes que el muro perdiera su capacidad de resistir y mostrara un alto degradamiento .

Resultado Experimental vs Modelo

Los resultados se muestran en la figura 3.6, donde se aprecia que en los primeros ciclos existe una mayor resistencia del modelo en la dirección positiva, en contraste con los desplazamientos negativos donde se ve una relación más cercana entre el muro y el modelo. Esto cambia posterior a los 20 [mm] y -20 [mm], donde el resultado comienza a ser similar al experimental en ambos sentidos. En los siguientes ciclos, y más en los finales, se ve un leve degradamiento y pérdida de simetría en el muro RW2. En el modelo crece la pendiente en el ciclo de descarga en comparación con el muro.

Se implementó un modelo de pandeo en el análisis (ver apéndice A), el cual no mostró diferencias con el análisis realizado con modelo de acero señalado en el capítulo 2, siendo consecuente con los resultados obtenidos experimentalmente al no presentar problemas de pandeo.

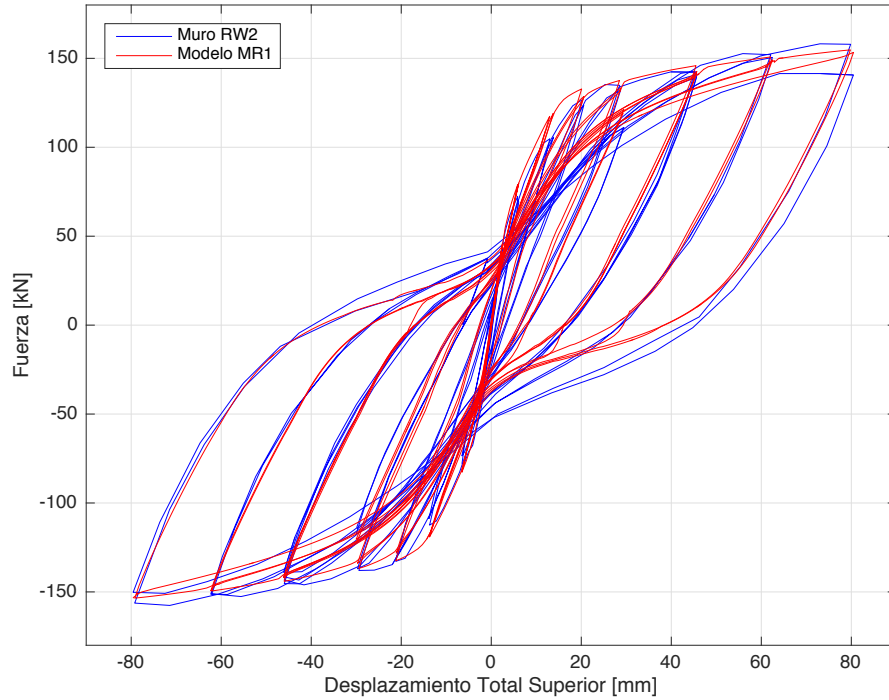


Figura 3.6: Gráfico de fuerza vs deformación total superior para muro experimental RW2 y modelo MR1

En el modelo MR1 se utilizaron coeficientes de daño de $\alpha_1 = 0.1$ y $\alpha_2 = 0.5$, sin presentar mayores problemas de convergencia y un resultado similar al del muro RW2, lo que se puede atribuir a la baja complejidad en comportamiento experimental del muro. El valor de $E_s = 200,000$ [MPa], aunque no se contó con los datos ensayados, mostró un correcto comportamiento de las barras de acero. Un aumento de f'_c del 10% ajusto mejor los resultados. El mallado se mantuvo con dimensiones dentro del rango 50 [mm] y 200 [mm], sin presentar problemas de convergencia e iguales dimensiones que el muro RW2.

Comparación de Envolvente

Al comparar las envolventes totales de ambas curvas obtenemos que inicialmente, el modelo posee una rigidez mayor en un 8%, según la tabla 3.6 y para el segundo punto de fluencia alcanza un 18% más, debido a las curvas en sentido positivo del ensayo. Posterior a estos puntos iniciales, las siguientes rigideces se mantienen dentro del 2% de diferencia.

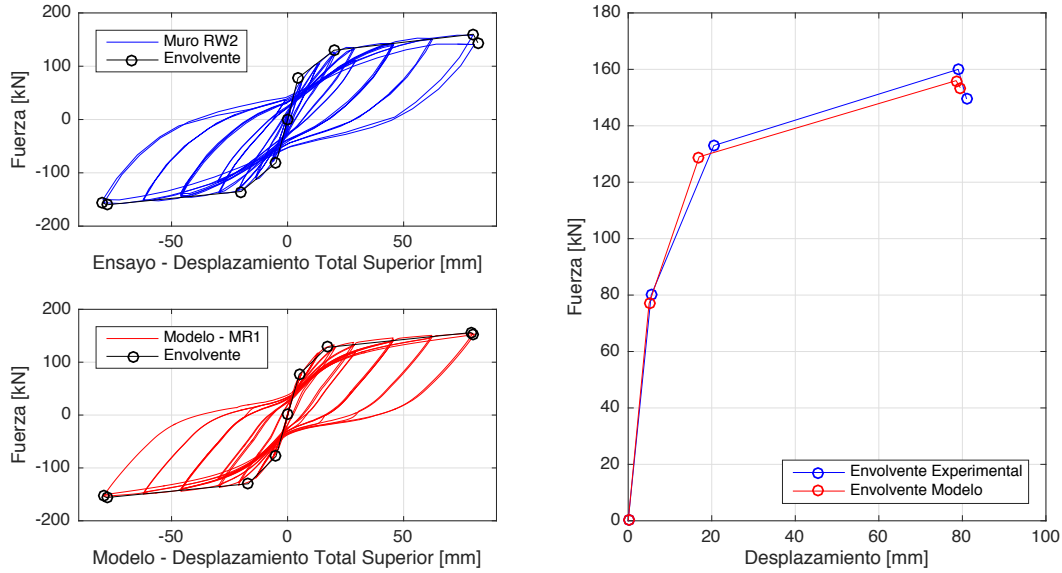


Figura 3.7: Izq: Envolvente total en muro RW2 y modelo MR1. Der: Promedio de envoltente en muro RW2 y modelo MR1

En las figuras 3.8 y 3.9 se compara el movimiento del primer piso, a una altura de 910 [mm]. Debido a que los nodos no calzan perfecto con la altura de los puntos de medición, se utiliza el promedio entre el nodo inferior (nodo 109) y superior (nodo 120), de modo de acercar el punto de medición al experimental. A partir de este resultado vemos un aporte sustancial de la flexión con respecto al corte para el movimiento del primer piso, correspondiente a casi un 50 % más por flexión que corte. Esto refleja el comportamiento del muro, siendo consecuente con la razón de aspecto de 3.0. En el Apéndice A se pueden ver los resultados de cada nodo, además del resultado utilizando la deformación angular del modelo.

En el desplazamiento por flexión se muestran pequeñas diferencias entre el modelo y el muro ensayado, no así en la deformación por corte, donde en los últimos ciclos dista por solo 1 [mm] más en el caso del modelo. El modelo logra ajustarse de correcta forma en el desplazamiento superior y en el caso de la flexión más que en el corte para el primer piso, representando el comportamiento del muro experimental.

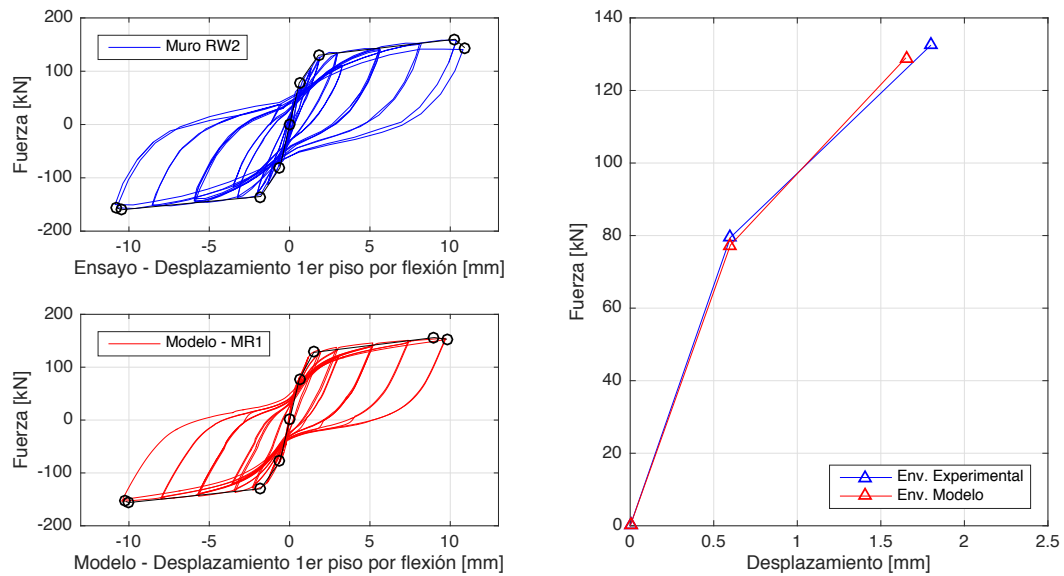


Figura 3.8: Izq: Envolvente por flexión en muro RW2 y modelo MR1. Der: Promedio de envolvente por flexión en muro RW2 y modelo MR1

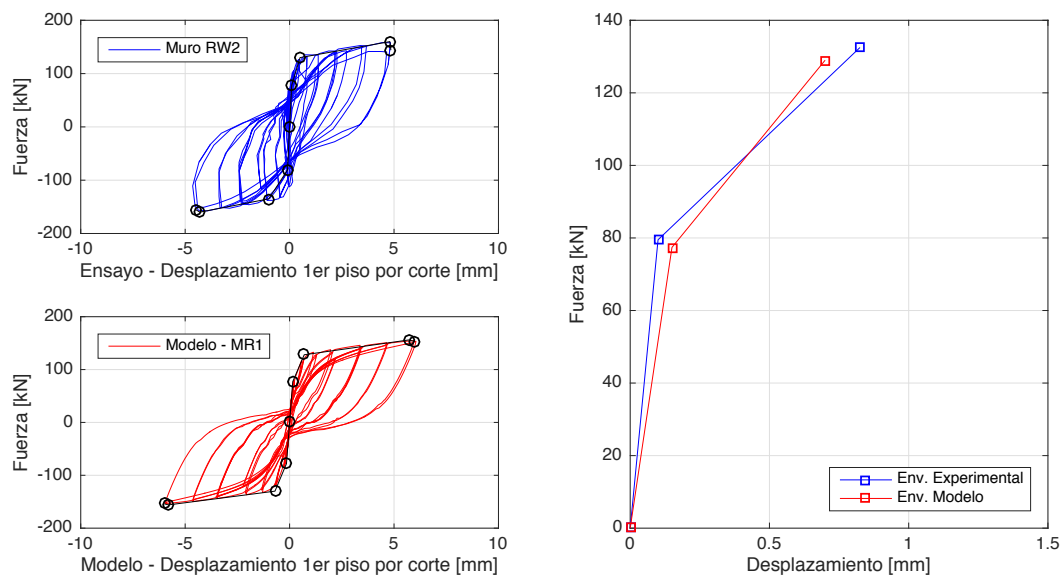


Figura 3.9: Izq: Envolvente por corte en muro RW2 y modelo MR1. Der: Promedio de envolvente por corte en muro RW2 y modelo MR1

Tabla 3.5: Parámetros de comparación - RW2 vs MR1

	Puntos en la Envolvente							
	Agrietamiento		Fluencia		$Corte_{max}$		$Desplazamiento_{max}$	
	$D_{tot,cr}$	V_{cr}	$D_{tot,Y}$	V_Y	$D_{tot,U}$	V_U	$D_{tot,X}$	V_X
Unidades	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]
Test	5.50	80	20.50	133	79.00	160	81.00	149.5
Modelo	5.00	77	17.00	129	78.50	156	79.50	153.5
Modelo/Test	0.91	0.97	0.83	0.97	0.99	0.98	0.98	1.03

Tabla 3.6: Comparación de Rigideces - RW2 vs MR1

	Rigidez					
	Inicial			Secante en Fluencia		
	$K_{ini,tot}$	$K_{ini,flex}$	$K_{ini,sh}$	$K_{Y,tot}$	$K_{Y,flex}$	$K_{Y,sh}$
Unidades	[kN/mm]	[kN/mm]	[kN/mm]	[kN/mm]	[kN/mm]	[kN/mm]
Test	14	133	795	6	74	168
Modelo	15	129	516	8	79	184
Modelo/Test	1.07	0.97	0.65	1.17	1.07	1.10

Tensión y Deformación

En las figuras 3.10 y 3.11 se señala la distribución de esfuerzos (S) y deformaciones (E) para un drift de 1 % y para d_{max} .

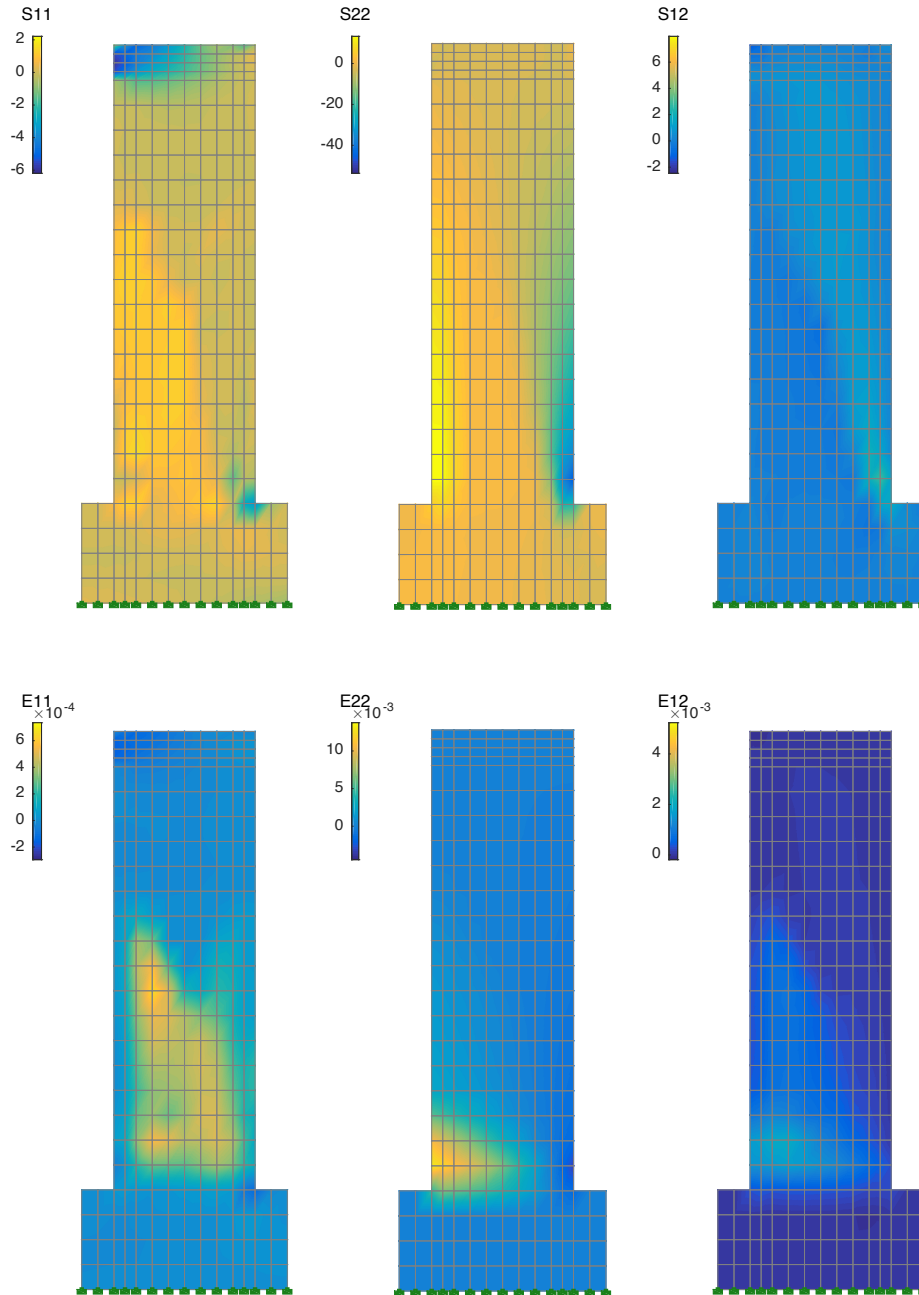


Figura 3.10: Tensión y deformación del modelo MR1 al 1 % de desplazamiento relativo

Las distribuciones de esfuerzos en ambas figuras poseen igual distribución, siendo consecuentes al presentar altas tensiones en el extremo superior para S11, además de aumentar los

esfuerzos para el desplazamiento máximo. Para S22 se obtienen esfuerzos marcados en la base del muro, debido a la compresión y tracción. finalmente, para S12 obtenemos un crecimiento de los esfuerzos de cizalle a lo largo del muro para el caso de d_{max} .

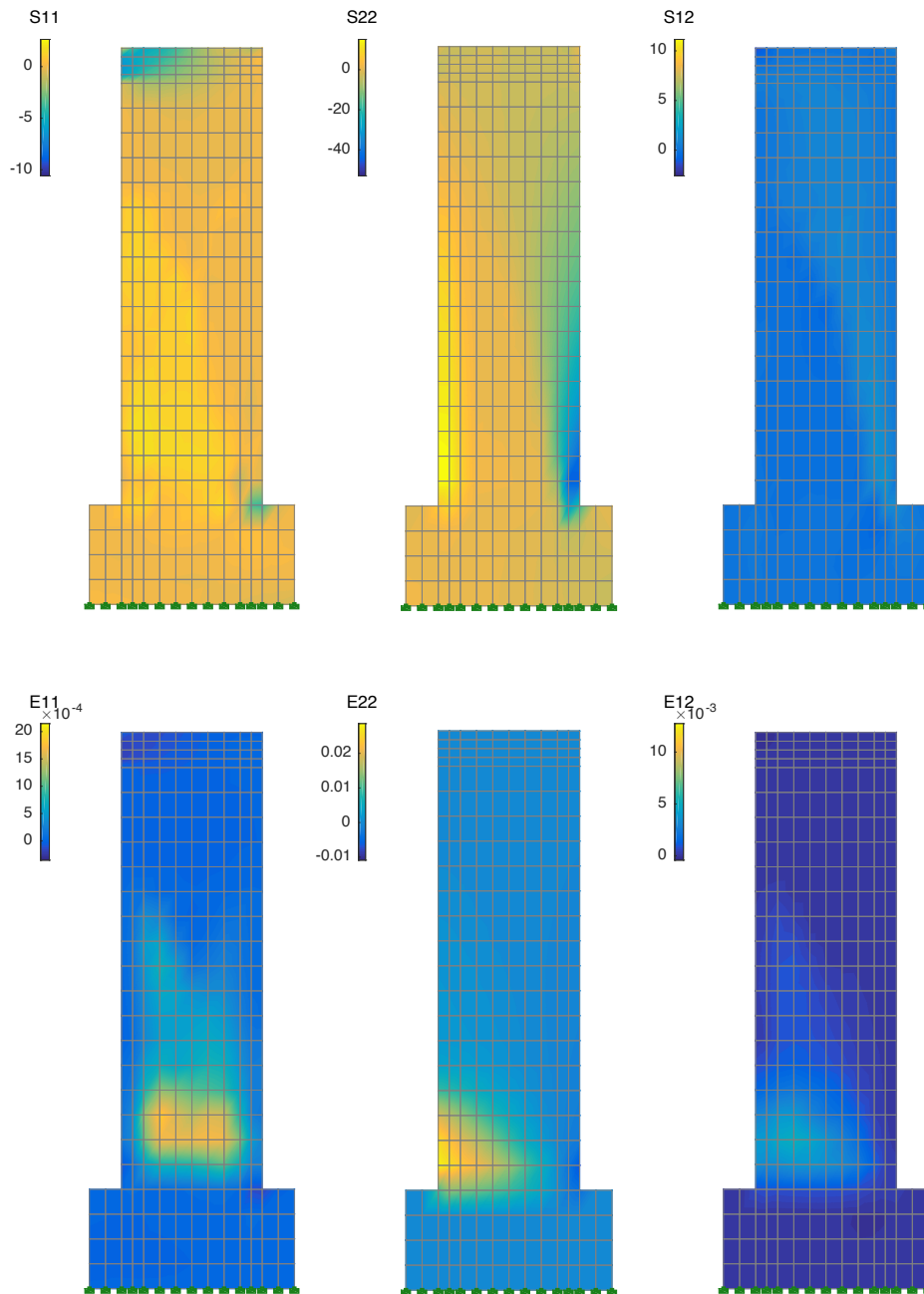


Figura 3.11: Tensión y deformación modelo MR1 al d_{max} - 2.21 % de deriva

En las deformaciones existe un aumento pronunciado en E11, concentrándose en la base para d_{max} . A su vez, en E22 se mantiene la distribución pero el cambio en la escala nos permite ver el aumento sustancial de las deformaciones. La deformación tangencial aumenta y se concentra en la zona traccionada del muro.

3.2. Caso 2 - MR2

3.2.1. Investigación: Experimental and Analytical Studies of Moderate Aspect Ratio Reinforced Concrete Structural Walls

Thien Anh Tran (2012) [14] investigó la no linealidad en la respuesta cíclica de muros de hormigón armado. Con este fin, se utilizaron 5 muros rectangulares en cantilever vertical, investigando tanto de forma experimental, como analítica la respuesta ante cargas cíclicas. Las variaciones entre los especímenes incluían cambios en el radio de aspecto (1.5 y 2.0), cantidad de carga axial ($0.025A_g f'_c$ y $0.10A_g f'_c$) y esfuerzos de corte en la base (entre 4 y $8\sqrt{f'_c}$ psi).

Todos los muros fueron ensayados por medio de un actuador axial superior que aplicaba una carga vertical, y una gato hidráulico dispuesto entre el muro y un muro de reacción, el cual ejercía las cargas cíclicas horizontales en la zona superior. A su vez, estos estaban contruidos sobre un pedestal, el cual se empotraba al suelo del laboratorio.

Los resultados de los experimentos mostraron que la pérdida de resistencia ante carga horizontal era producto de la tensión y compresión en la diagonal del muro, cizalle en fracturas y aplastamiento del hormigón y del refuerzo vertical.

Utilizando dos acercamientos para modelar los muros (P-M y V acoplados y no acoplados), se concluyó que ambos modelos sobrestiman la capacidad a carga lateral y su rigidez ante desplazamientos laterales. Finalmente, se dan bases para realizar un modelo más preciso de los muros para cargas cíclicas y su posible expansión a modelos no-lineales.

Se utiliza el espécimen RW-A15-P10-S78 para este trabajo, modelo al cual se le asigna el nombre MR2.

3.2.2. Muro de Estudio: RW-A15-P10-S78

Este espécimen es un muro de hormigón armado rectangular, de razón de aspecto de 1.5, apoyado sobre un pedestal de fundación y sometido a una carga superior horizontal de forma cíclica. Además de una carga vertical estática de 641 [kN], es decir, $P/(A_g \cdot f'_c) = 0.10$, cargados en la zona superior, compuesta de igual ancho que el muro pero mayor refuerzo. El muro tuvo un comportamiento de muro de corte, mostrando una falla horizontal en la conexión pedestal-muro. Posteriormente, falló debido a la compresión excesiva de forma diagonal en la zona inferior y altos esfuerzos de cizalle. En la figura 3.12 se puede ver el montaje experimental del muro RW-A15-A10-S78.

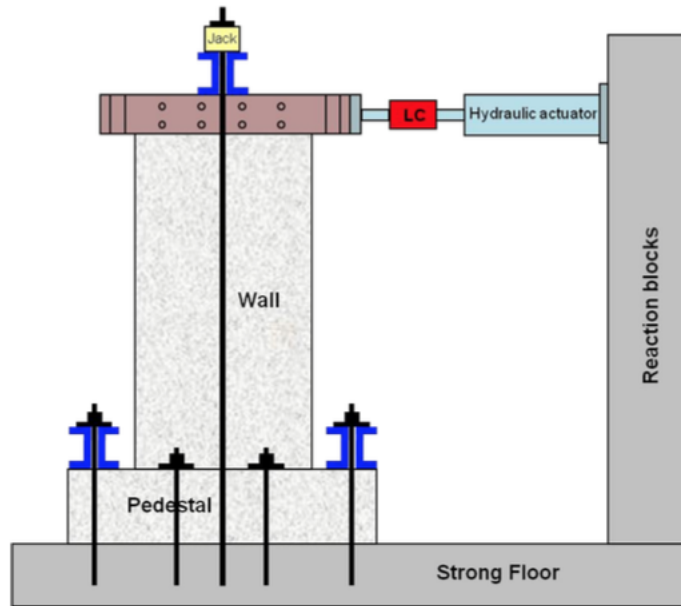


Figura 3.12: Montaje experimental ensayado al muro RW-A15-P10-S78. (Imagen: Tran, 2012 [14])

Dimensiones y Geometría

La altura, ancho y espesor del muro se especifican en la tabla 3.7, junto a otras dimensiones relevantes. En la figura 3.13 se puede apreciar el muro en proyección.

Tabla 3.7: Geometría Muro RW-A15-P10-S78

Dimensión	Valor
$H_w [mm]$	1828
$L_w [mm]$	1220
L_{conf}	210
$T_w [mm]$	150
H_w/L_w	1.50
L_w/T_w	8.13
s/d	2.67

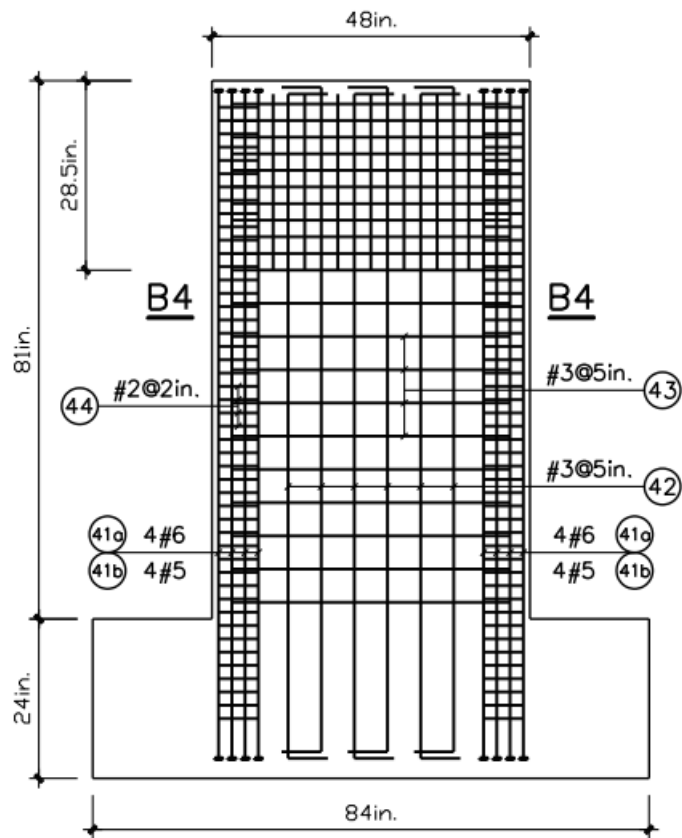


Figura 3.13: Dimensiones y geometría del muro RW-A15-P10-S78. (Imagen: Tran, 2012 [14])

Armadura de refuerzo

Las cuantías de refuerzo del muro, en porcentaje, se señalan en la tabla 3.8. La sección transversal del muro, incluyendo la armadura de refuerzo, se muestra en la figura 3.14.

Tabla 3.8: Cuantías de refuerzo en Muro A15-P10-S78

Cuantía	[%]
ρ_v [%]	0.71
ρ_h [%]	0.63
$\rho_{estribos}$ [%]	0.83
ρ_{conf} [%]	6.15

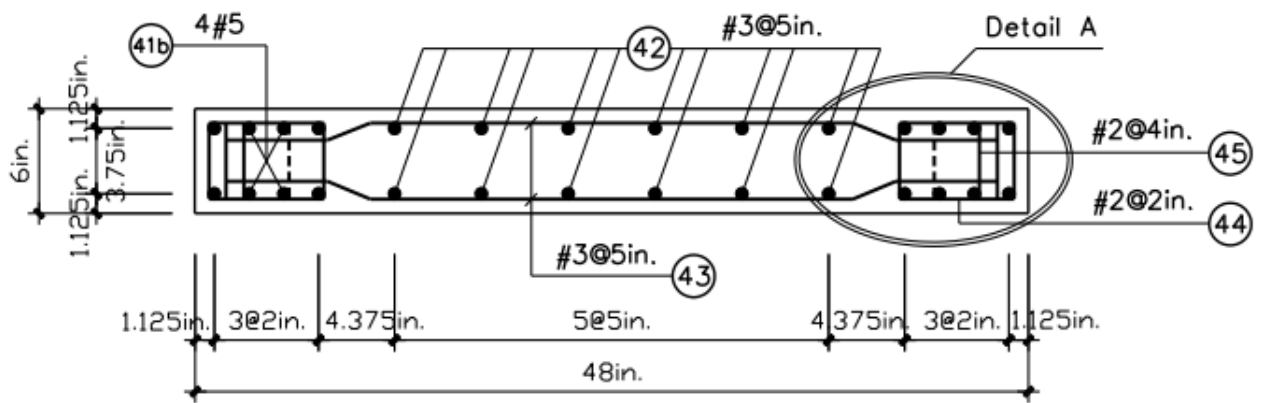


Figura 3.14: Sección transversal del muro RW-A15-P10-S78. (Imagen: Tran, 2012 [14])

Materiales

Para cada uno de los materiales, tanto barras de acero como hormigón, se especifican las capacidades de resistencia y módulo de Young utilizados para el diseño y ensayos para el muro A15-P10-S78, en la tabla 3.9.

Tabla 3.9: Características de materiales

Característica	Diseño	Experimental
f'_c [MPa]	34.5	55.8
f_{yv} [MPa]	414	443
f_{yh} [MPa]	414	443
f_{yconf} [MPa]	414	477
f_{ys} [MPa]	276	423
E_s [MPa]	200,000	[-]*

*: La investigación o reporte de los ensayos no presentó datos del módulo de Young calculado experimentalmente.

Cargas aplicadas

La carga vertical estática correspondía a $641[KN]$, o $P/(A_g \cdot f'_c) = 0.06$, aplicada en la zona superior en forma distribuida. La carga horizontal fue aplicada de forma cíclica, por medio de un actuador hidráulico. Posterior al agrietamiento, el ensayo fue controlado por su desplazamiento relativo Δ . En la figura 3.15 se muestra el protocolo de carga y desplazamiento al que fue sometido el muro experimental, del cual se utilizarán los mismos desplazamientos para comparar los resultados entre el modelo y el muro experimental.

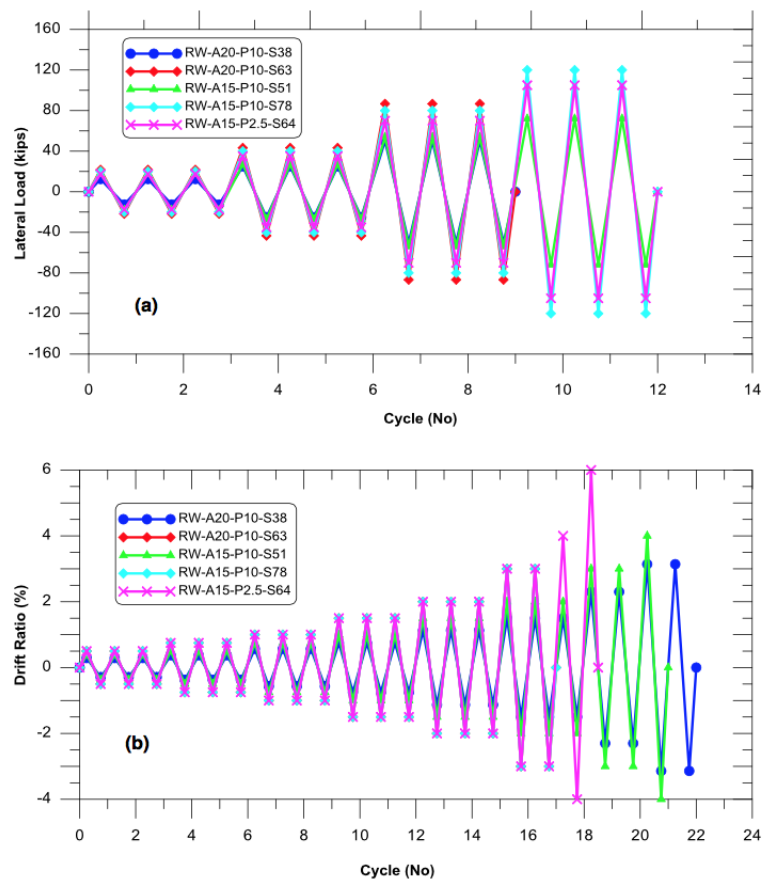


Figura 3.15: Protocolo de deformación para ensayo del muro RW-A15-P10-S78. (Imagen: Tran, 2012 [14])

3.2.3. Modelo

Mallado

El muro se modeló por medio de 8 elementos horizontales, de los cuales dos elementos a cada extremo representaban la zona confinada, con anchos de 100 [mm] y 110 [mm], y otros 4 elementos centrales de ancho 200 [mm]. Verticalmente consta de 9 elementos de altura 203 [mm], seguidos por otros 4 elementos de altura 57 [mm] que representan la viga de carga. El pedestal de fundación fue constituido por cuatro elementos verticales de altura 203 [mm] y 12 elementos horizontales de ancho similar al dispuesto en el muro. En la figura 3.16 se puede ver el mallado de forma gráfica.

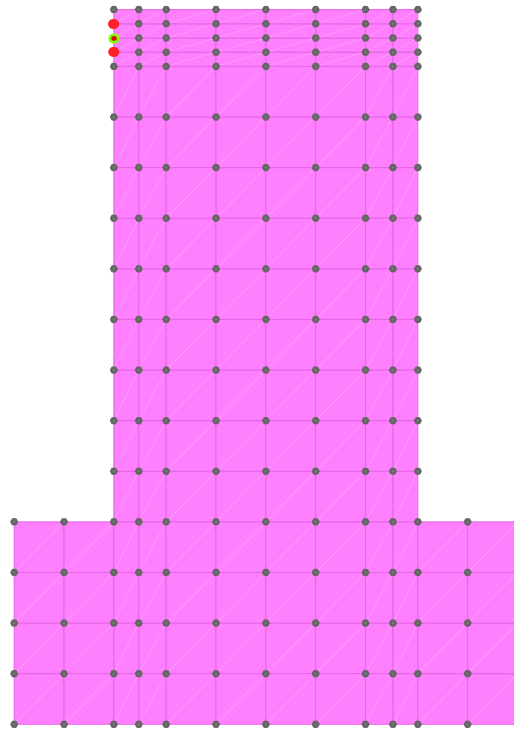


Figura 3.16: Partición Muro MR2

Cargas

Las fuerzas actuantes sobre el modelo se ejercieron en los seis nodos centrales superiores con una carga de $-641 [kN]$ distribuidos en los nodos, y las cargas horizontales fueron aplicadas en tres nodos izquierdos en la viga de carga, señalados en rojo en la figura 3.16. En verde, el nodo de registro de desplazamientos.

Coefficientes de daño

Los coeficientes de daño utilizados, por sector, se señalan en la tabla 3.10.

Tabla 3.10: Coeficientes de modelo - Modelo MR2

Coeficiente	Valor
α_{1conf}	0.1
α_{2conf}	0.4
α_1	0.1
α_2	0.4
K_{conf}	1.05

3.2.4. Análisis y Resultados

El muro RW-A15-P10-S78 tuvo un comportamiento más al corte que a la flexión, lo cual se relaciono en su investigación a su razón de aspecto de 1.5. Con una relación $s/d=2.67$, no se obtuvieron problemas de pandeo en sus barras confinadas. Su ensayo culmino tras graves efectos de compresión diagonal y esfuerzos de cizalle graves en su zona inferior.

Resultado Experimental vs Modelo

El modelo MR2 fue modelado con con menos elementos que el modelo MR1, su resultado se muestra en la figura 3.17. Los coeficientes de daño utilizados fueron de $\alpha_1 = 0.1$ y $\alpha_2 = 0.4$, los cuales permitieron ajustar el comportamiento del modelo. El mallado realizado fue con menos cantidad de elementos que los demás modelos debido a sus dimensiones.

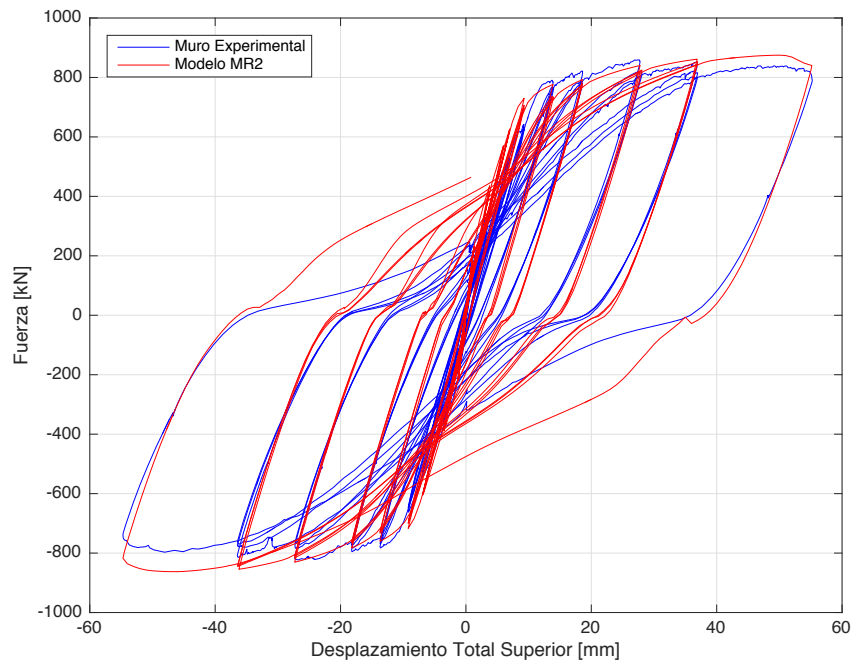


Figura 3.17: Ciclo Carga-Desplazamiento MR2 y A15-P10-S78

En los resultados de la figura 3.17 se aprecian primeros ciclos de mayor resistencia para el modelo, lo cual cambia al superar un desplazamiento de $10 [mm]$ y $-10 [mm]$, en donde el resultado comienza a ser similar al experimental. Sobre un drift de 1.5% ($\sim 30 [mm]$) e incrementándose en los ciclos finales, se ve un leve degradamiento del muro ensayado y simetría en el muro modelado MR2. En cuanto al ciclo de descarga, sobre $20 [mm]$ de desplazamiento, el modelo muestra una curva que difiere del ensayo para los valores de fuerza cercanos a 0. Esta respuesta se relaciona con un error numérico, dado que no afecta la capacidad del modelo de representar la respuesta en su ciclo de carga y capacidad máxima. A su vez, el modelo presenta simetría tanto para el desplazamiento positivo, como negativo, lo cual no se

presenta en la respuesta experimental, el cual muestra leves disminuciones en la resistencia máxima al realizar el desplazamiento en sentido negativo.

Comparación de Envolvente

En la envolvente total se aprecia una mayor capacidad del modelo con respecto al muro experimental. Inicialmente, la rigidez es similar en ambos casos, comenzando a distanciarse en el punto de fluencia, donde el muro es levemente más flexible. Como se aprecia en la figura 3.18, en los puntos de carga máxima y desplazamiento máximo, el modelo se mantiene similar forma pero mayor resistencia que el muro experimental.

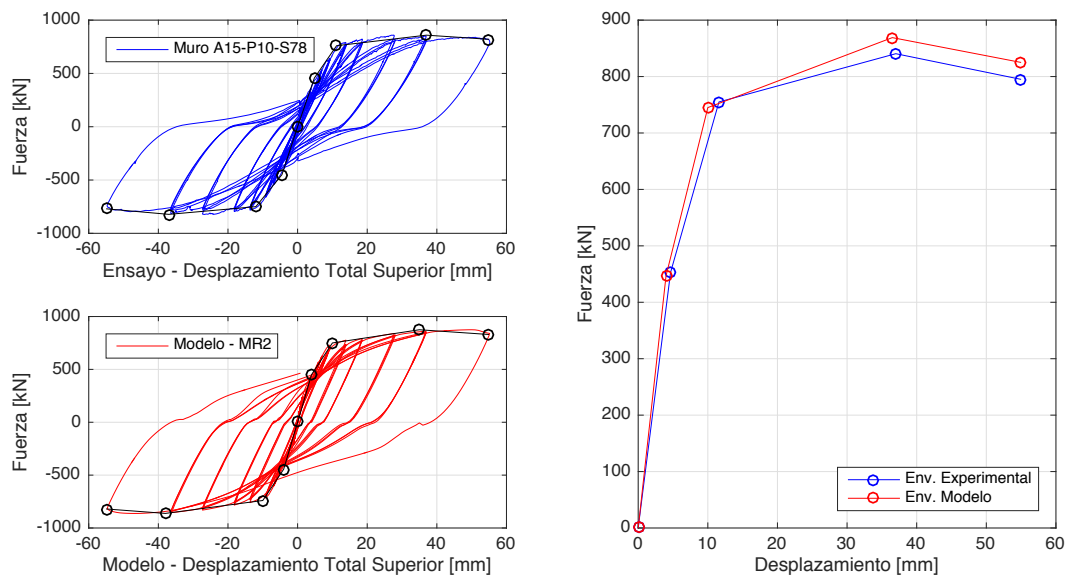


Figura 3.18: Izq: Envolvente total en muro RW-A15-P10-S78 y modelo MR2. Der: Promedio de envolvente en muro RW-A15-P10-S78 y modelo MR2

En el desplazamiento por flexión, figura 3.19, las curvas para la máxima capacidad difieren en el último ciclo negativo, donde el muro experimental no logra simetría con respecto al valor alcanzado en dirección positiva. En cuanto las rigidez inicial y de fluencia muestra al modelo más rígido que el muro experimental.

En el caso del aporte por corte, figura 3.20, el modelo muestra mayores desplazamientos que el muro experimental para un desplazamiento sobre los 5 [mm], además de diferencias de simetría en el muro experimental. En cuanto a la rigidez inicial como de fluencia, figura 3.20 a la derecha, el modelo se ajusta al muro experimental. En este modelo, tanto el desplazamiento por corte como el desplazamiento por flexión es de similar magnitud, siendo consecuente con su razón de aspecto.

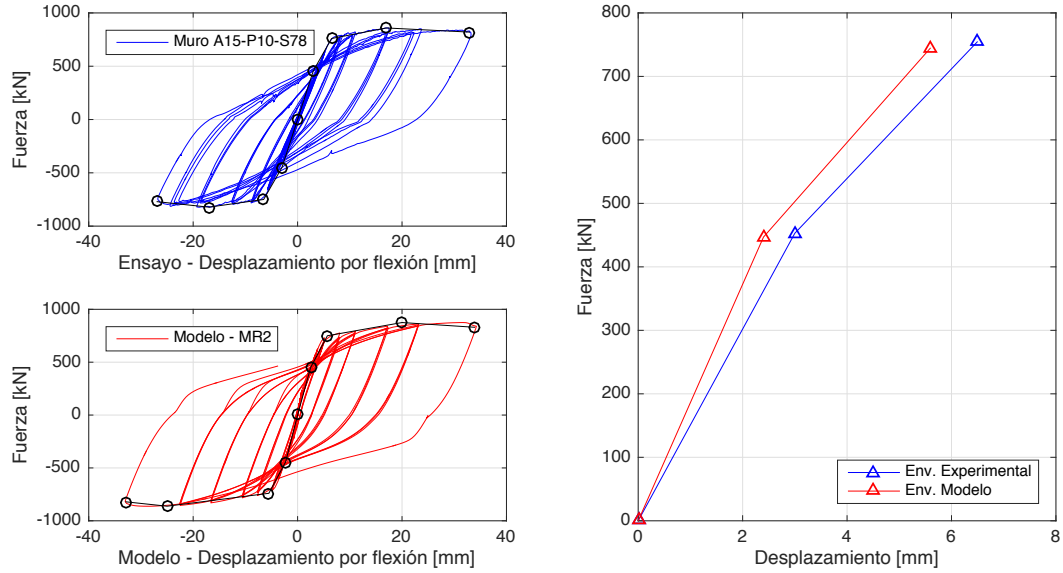


Figura 3.19: Izq: Envolvente por flexión en muro RW-A15-P10-S78 y modelo MR2. Der: Promedio de por flexión en muro RW-A15-P10-S78 y modelo MR2

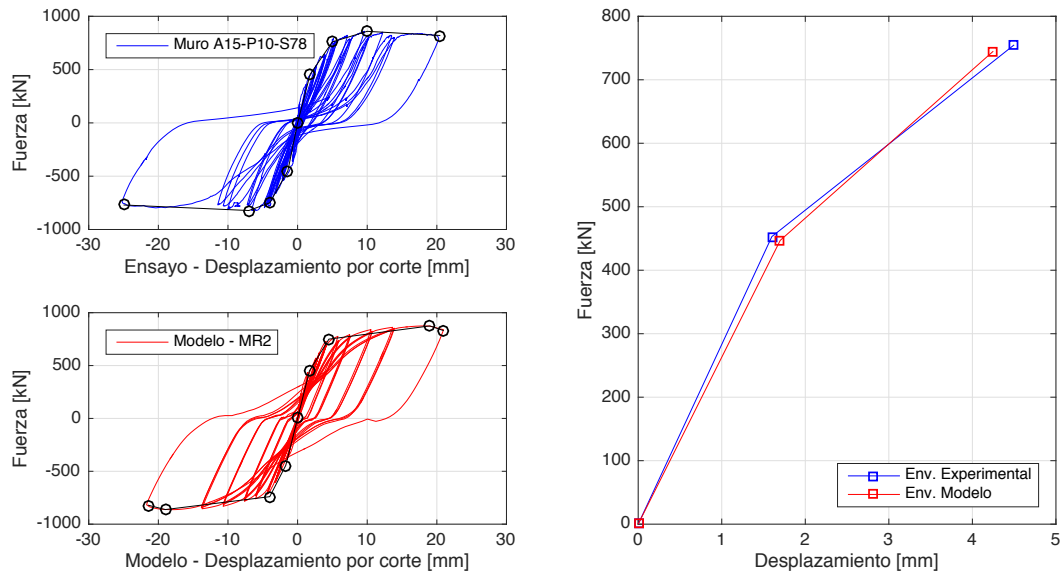


Figura 3.20: Izq: Envolvente por flexión en muro RW-A15-P10-S78 y modelo MR2. Der: Promedio de por flexión en muro RW-A15-P10-S78 y modelo MR2

Los resultados corresponden a los obtenidos utilizando el método experimental por giro, revisado en el capítulo 2. En el Apéndice A se pueden ver los resultados utilizando la deformación angular del modelo.

Tabla 3.11: Parámetros de comparación - RW-A15-P10-S78 vs MR2

	Puntos en la Envolvente							
	Agrietamiento		Fluencia		$Corte_{max}$		$Desplazamiento_{max}$	
	$D_{tot,cr}$	V_{cr}	$D_{tot,Y}$	V_Y	$D_{tot,U}$	V_U	$D_{tot,X}$	V_X
Unidades	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]
Test	4.65	453	11.60	755	37.00	841	55.00	795
Modelo	4.00	447	10.00	745	36.50	869	55.00	825
Modelo/Test	0.86	0.99	0.86	0.99	0.99	1.03	1.00	1.04

Tabla 3.12: Comparación de Rigideces - RW-A15-P10-S78 vs MR2

	Rigidez					
	Inicial			Secante en Fluencia		
	$K_{ini,tot}$	$K_{ini,flex}$	$K_{ini,sh}$	$K_{Y,tot}$	$K_{Y,flex}$	$K_{Y,sh}$
Unidades	[kN/mm]	[kN/mm]	[kN/mm]	[kN/mm]	[kN/mm]	[kN/mm]
Test	98	151	287	65	116	170
Modelo	112	187	263	75	133	176
Modelo/Test	1.15	1.24	0.91	1.14	1.15	1.04

Tensión y Deformación

En las figuras 3.21 y 3.22 se señala la distribución de esfuerzos (S) y deformaciones (E) para un drift de 1 % y para d_{max} .

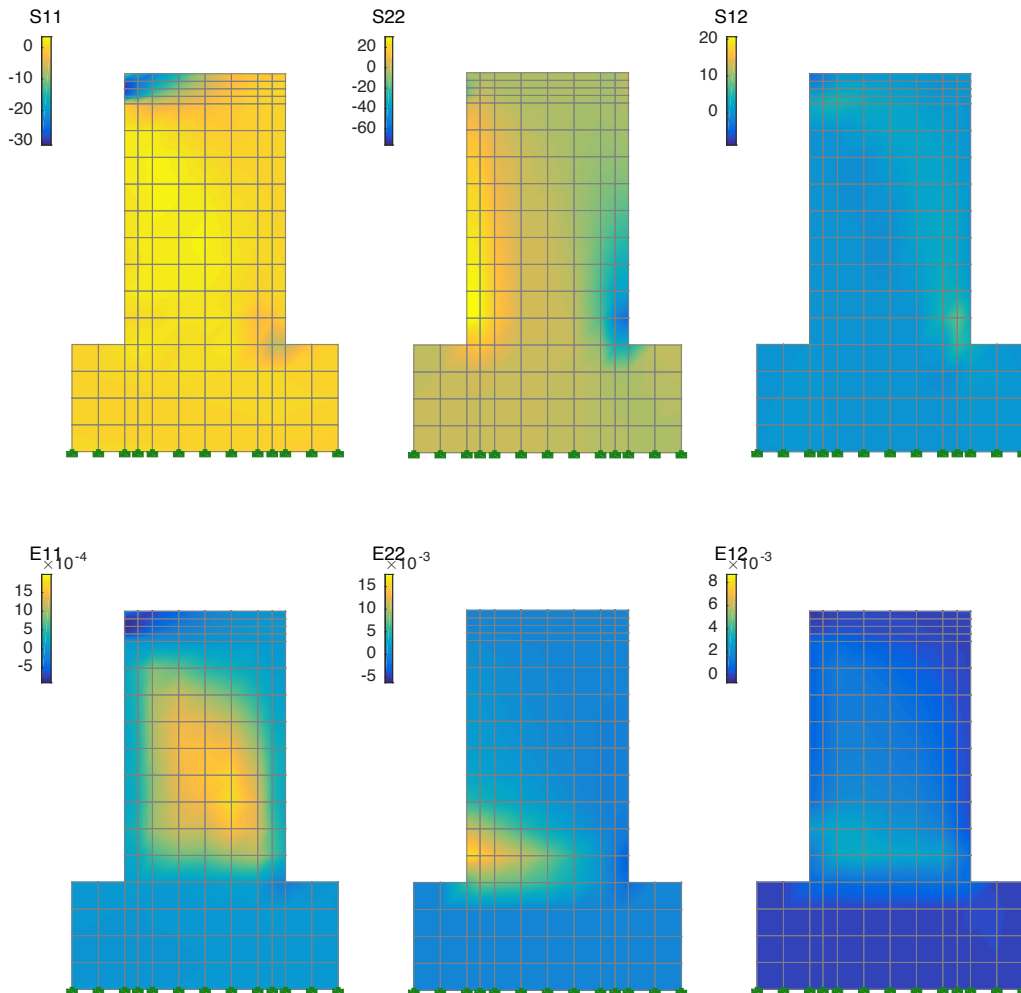


Figura 3.21: Tensión y deformación modelo MR2 al 1 % de desplazamiento relativo

En los esfuerzos podemos ver bastante similitud entre ambos estados de carga. Los esfuerzos de S11, S22 y S12 son consecuentes con la carga positiva, manteniendo altos esfuerzos en el extremo donde actúa la carga para S11, y en los extremos inferiores para S22, al estar sujetos a compresión y tracción.

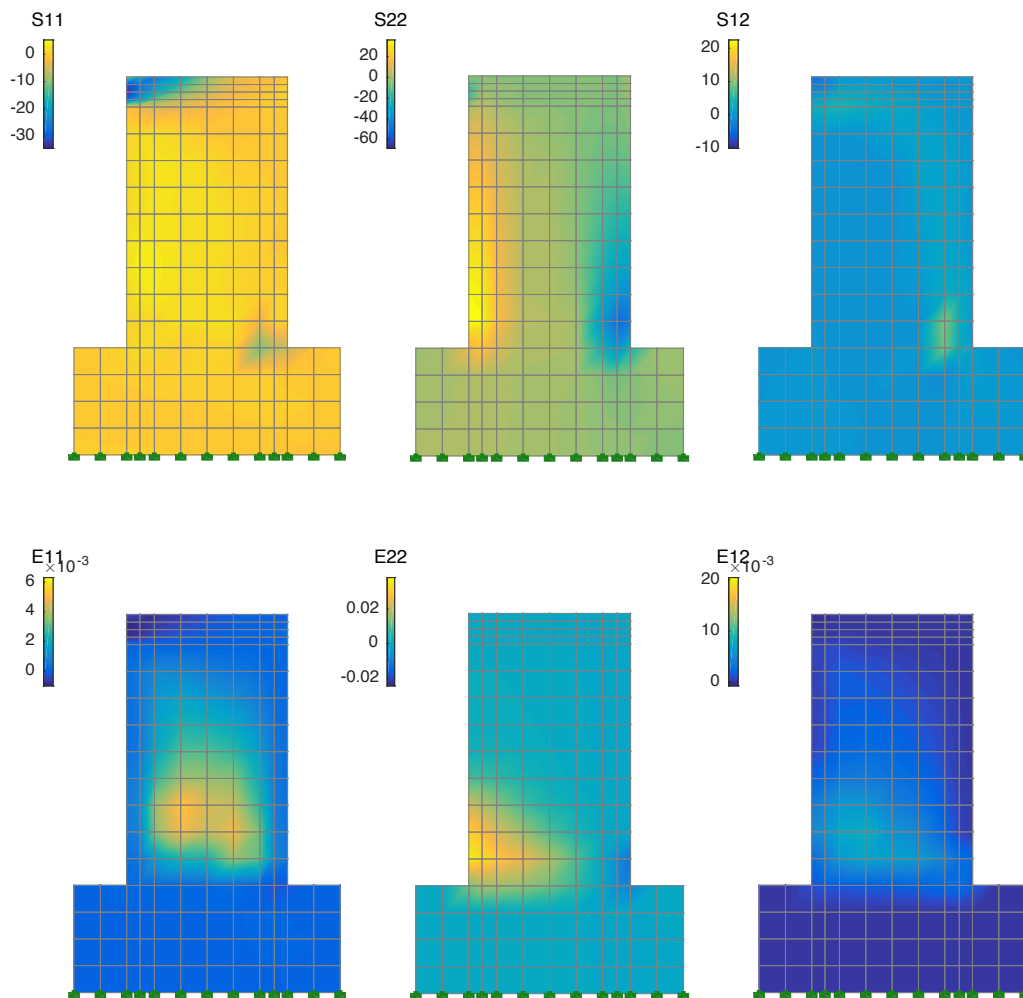


Figura 3.22: Tensión y deformación modelo MR2 al d_{max} - 3.01% de deriva

Los cambios más notorios se aprecian en la deformación, la cual aumenta considerablemente para d_{max} , concentrando las deformaciones en la parte inferior para E11. Existe una gran deformación por tracción y compresión en la base para la deformación axial E22. A su vez, en E12 la deformación en d_{max} alcanza a duplicar el caso de drift 1%, concentradas en la zona inferior.

3.3. Caso 3 - MR3

3.3.1. Investigación: Quasi-static Cyclic Tests and Plastic Hinge Analysis of RC Structural Walls

Dazio, A., Beyer, K. y Bachmann, H., (2009) [3] realizaron una serie de ensayos en el Instituto de Ingeniería Estructural (IBK), en la Escuela Politécnica Federal de Zúrich, Suiza.

El objetivo de esta investigación fue estudiar la influencia del porcentaje de armadura dispuesta en la ductilidad de los muros, para cuantías comúnmente utilizada en muros rectangulares en Europa. Este porcentaje varía entre 2 a 5% en países con alta sismicidad y ronda el 1% en países con sismicidad media.

Los ensayos realizados consistieron en la aplicación de cargas cíclicas cuasiestáticas a seis muros de hormigón armado. La geometría y la armadura horizontal era igual para todos los muros, sin embargo, la cantidad y disposición de la armadura vertical y la armadura del confinamiento variaba según muro.

Los muros fueron construidos sobre un pedestal de fundación de base 2800[mm], altura 600[mm] y profundidad 700[mm]. La zona superior de los muros correspondía a una viga de carga, y expandía a 400[mm] de ancho y mayor cuantía de refuerzo, zona donde se aplicaba la carga horizontal y vertical.

Se utilizaron dos actuadores hidráulicos, en serie, con una capacidad de 500 [kN] y ± 100 [mm]. Para los primeros 5 muros y un actuador hidráulico de 1000 [kN] y ± 100 [mm] para el sexto muro, el cual ejercería la carga horizontal cíclica. La carga axial vertical se aplicó por medio de dos tensores dispuestos en la cara superior de los muros. El comportamiento de los muros fue medido por sensores. Además, cada ciertos intervalos se realizaba una medición manual. Se registraron los desplazamientos globales, locales y fuerzas aplicadas. Además, se realizaron mediciones manuales de abertura de grietas.

Las conclusiones de la investigación son las siguientes:

- Muros con bajo refuerzo longitudinal poseen normalmente menor cantidad de fisuras. Esto implica que la tensión resultante por flexión también es baja en esos muros. Como resultado de esto, se obtiene una concentración de deformaciones plásticas en la base, disminuyendo su capacidad de desplazamiento.
- Es posible tener grietas más grandes en el centro que en los extremos confinados de los muros si se cumple: un patrón de agrietamiento, poca o nula ductilidad de la armadura del centro del muro y buenas uniones entre los elementos del muro.
- Normalmente, la primera fractura del muro cruza todo el muro. Además, el aumento de los esfuerzos de los elementos de borde con respecto al desplazamiento superior no es proporcional a lo largo del ensayo, y es la causal de la falla la totalidad del muro.
- Los ensayos demuestran que se produce una disminución de la capacidad de deformación y ductilidad ante menor cuantía de armadura vertical, lo cual se ve incrementado al usar barras con menos capacidad de elongación.

Para modelar y comparar los resultados obtenidos, se escoge el muro sexto de la investigación, el espécimen MSH6.

3.3.2. Espécimen de Estudio: WSH6

El muro WSH6 es un muro de hormigón armado rectangular con ambos extremos confinados, apoyado sobre un pedestal, y con una viga de carga en la zona superior de mayor cuantía de armadura y mayores dimensiones donde se ejercen las fuerzas y se miden las deformaciones. Este muro es cargado verticalmente de forma estática por 1,476 [kN], lo que equivale a $P/(A_g \cdot f'_c) = 0.11$. A su vez, es cargado horizontalmente de forma cíclica, en su extremo superior, realizando ciclos de carga controlados por la deriva superior y aumentando hasta la falla del muro.

Dimensiones y Geometría

La altura, ancho y espesor del muro se especifican en la tabla 3.13, junto a otras dimensiones relevantes. En la figura 3.23 se muestra el muro en proyección.

Tabla 3.13: Geometría Muro WSH6

Dimensión	Valor
$H_w [mm]$	4030
$L_w [mm]$	2000
$L_{conf} [mm]$	385
$T_w [mm]$	150
H_w/L_w	2.02
L_w/T_w	13.33
s/d	4.17

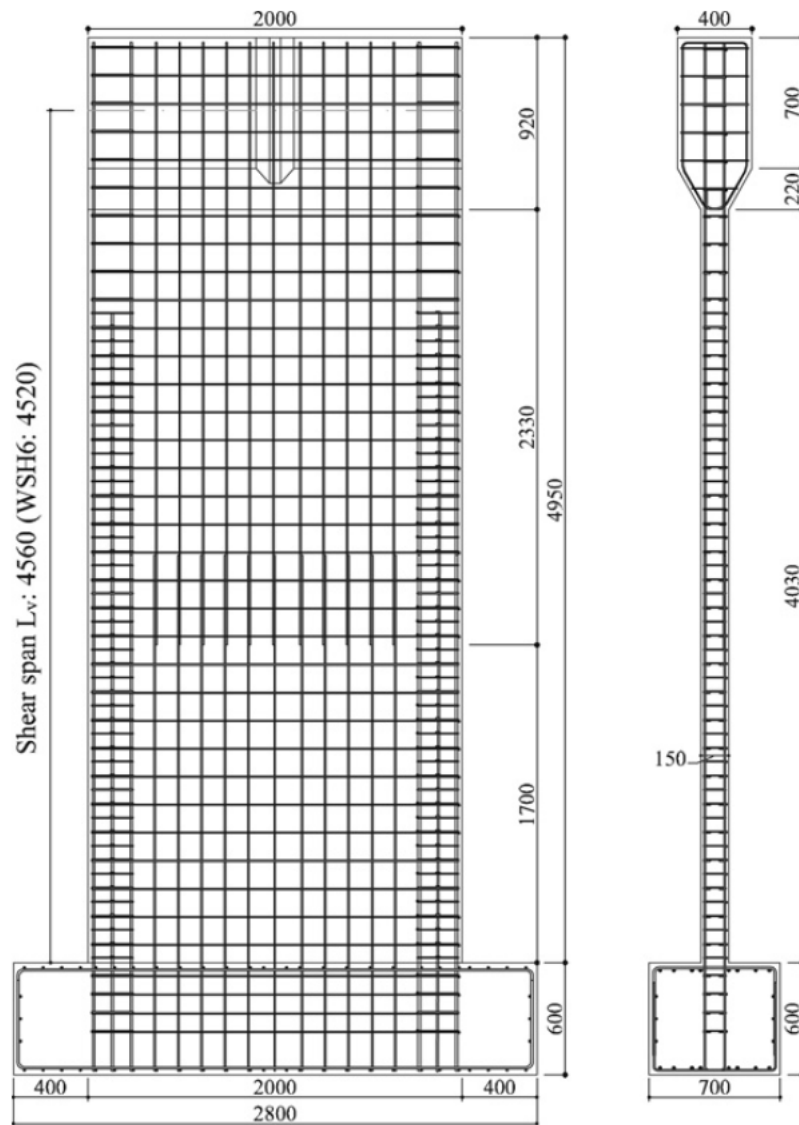


Figura 3.23: Dimensiones del muro WSH6. (Imagen: Dazio, 2009 [3])

Armadura de refuerzo

En la tabla 3.14 se presentan las cuantías de refuerzo (%), según sección del muro. La disposición de los estribos y barras se pueden ver en la figura 3.24.

Tabla 3.14: Cuantías de refuerzo en Muro WSH6

Cuantía	ρ [%]
Vertical [%]	0.54
Horizontal [%]	0.25
Estribos [%]	0.88
Conf. Vertical [%]	1.35

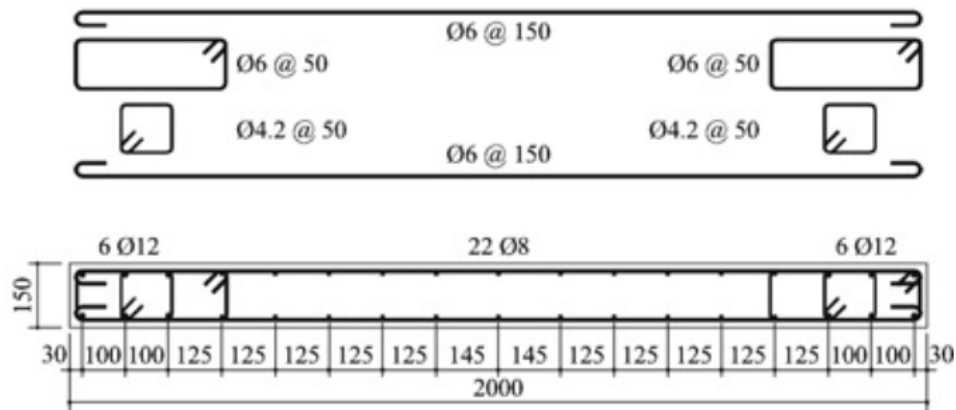


Figura 3.24: Disposición de armadura en el muro WSH6. (Imagen: Dazio, 2009 [3])

Materiales

Para cada uno de los materiales, tanto barras de acero como hormigón, se especifican las capacidades de resistencia y módulo de Young utilizados para el diseño y obtenidos experimentalmente para el muro WSH6, en la tabla 3.15.

Tabla 3.15: Características de materiales

Característica	Diseño	Experimental
f'_c [MPa]	45	45.6
f_{yv} [MPa]	546	540
f_{yh} [MPa]	486	502
f_{yconf} [MPa]	535	542
f_{ys} [MPa]	535	543
E_s [MPa]	200,000	210,300

Cargas aplicadas

La carga vertical estática aplicada corresponde a $1,476[kN]$, o $P/(A_g \cdot f'_c) = 0.11$. La carga horizontal se aplicaba de forma cíclica, determinada por el protocolo de carga mostrado en la figura 3.25, según la deriva superior del muro.

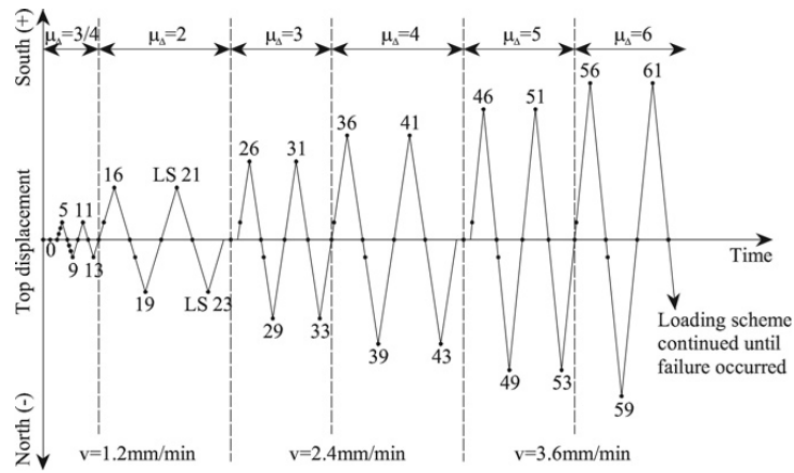


Figura 3.25: Protocolo de deformación para ensayo de muro WSH6. (Imagen: Dazio, 2009 [3])

3.3.3. Modelo

Mallado

El muro se modeló con 12 elementos horizontales, 3 elementos a cada extremo representa la zona confinada, con anchos de 130, 130 y 125 [mm], y otros cuatro elementos centrales de ancho 205 [mm]. La partición vertical consta de 17 elementos de altura 237 [mm]. El pedestal de fundación está constituido por cuatro elementos verticales de altura 237 [mm] y 16 elementos horizontales de ancho similar al dispuesto en los elementos de muro adyacentes. La viga de carga consistió en 4 elementos verticales dispuestos sobre el muro, de 230 [mm] de alto y ancho similar al dispuesto en el muro. En la figura 3.26 se puede ver la malla de forma gráfica.

Cargas

Las fuerzas verticales actuantes sobre el modelo se ejercieron en los 10 nodos centrales de la viga de carga, con una carga de $-147,600[N]$ por nodo, y las cargas horizontales fueron aplicadas en los tres nodos izquierdos señalados en rojo en la figura 3.16. En verde, el nodo de registro de desplazamientos.

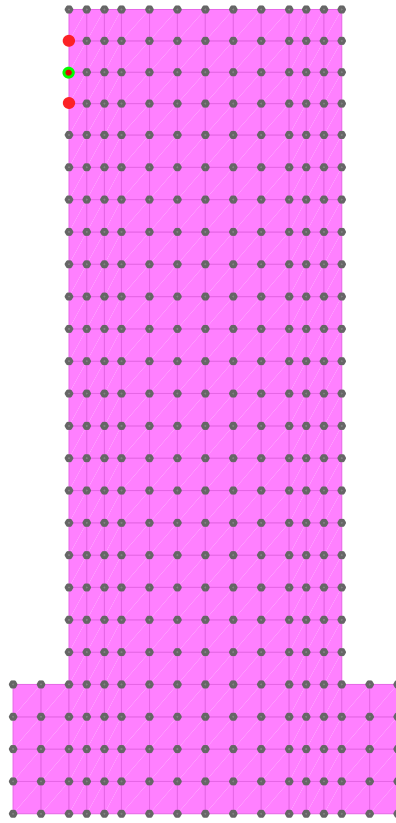


Figura 3.26: Partición Muro MR3

Coefficientes del modelo

Los coeficientes de daño utilizados se señalan en la tabla 3.16.

Tabla 3.16: Coeficientes de modelo - Modelo MR3

Coeficiente	Valor
α_{1conf}	0.1
α_{2conf}	0.4
α_1	0.1
α_2	0.4
K_{conf}	1.13

3.3.4. Análisis y Resultado

El muro experimental WSH6, de razón de aspecto 2.02, mostraba un resultado experimental gobernado por la flexión, empezando a degradar posterior a los 80 [mm] de desplazamiento. A su vez, claros patrones de agrietamiento y una zona confinada de gran ancho (385 [mm]).

Resultado Experimental vs Modelo

El modelo incorporó un mallado más fino que los demás modelos en su zona confinada, conteniendo 3 elementos debido a la magnitud de esta zona, incluyendo diferentes cuantías en cada parte, representando la disposición del muro experimental. Utilizando los valores de α_1 y α_2 indicados en la tabla 3.16 y con un mallado de anchos y altos de elemento cercanos a los 200[mm], tres elementos en la zona confinada de aproximadamente 100 [mm], replicando al muro experimental WSH6, se obtuvo el resultado de la figura 3.27. Los resultados presentan un comportamiento del modelo menos resistente que el muro experimental, sin embargo, logra predecir su comportamiento a lo largo de los ciclos y su punto de degradamiento.

Además de este resultado, se implemento un modelo de pandeo en el análisis, el cual no mostró diferencias con el análisis realizado con modelo de acero señalado en el capítulo 2, siendo consecuente con los resultados obtenidos experimentalmente al no presentar problemas de pandeo.

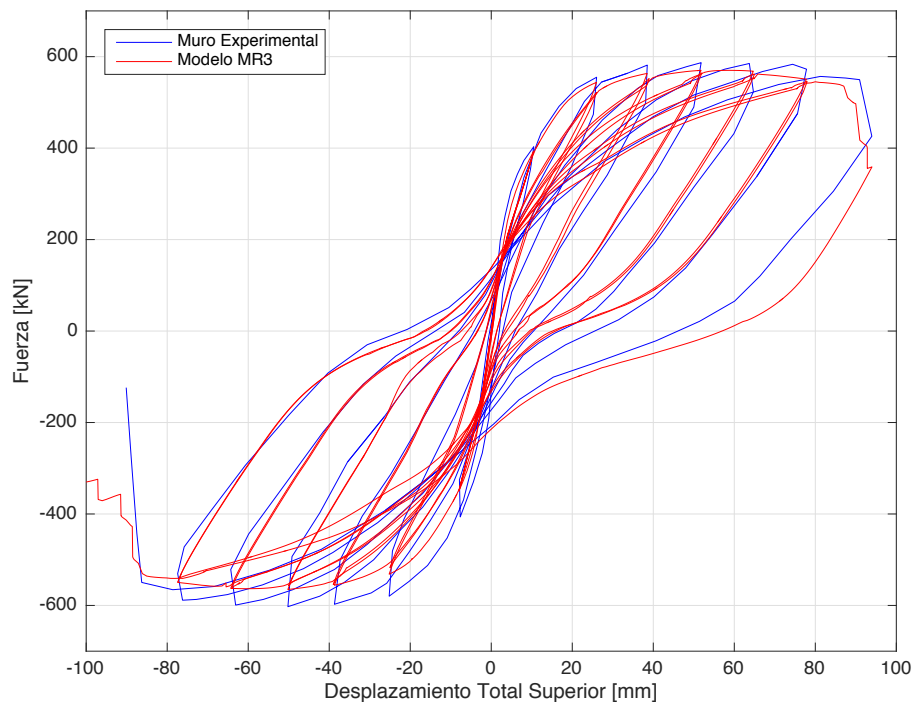


Figura 3.27: Gráfico de fuerza vs deformación total superior para muro experimental WSH6 y modelo MR3

Comparación de Envolvente

A partir de los puntos de la envolvente y la rigidez de la figura 3.28, se denota un correcto comportamiento del modelo, replicando el comportamiento del muro experimental, obteniendo un decaimiento en puntos similares aunque menos resistencia que el muro experimental.

Es posible ver un aumento de la resistencia del muro experimental en el sector de los desplazamientos negativos, lo cual el modelo logra representar. En el muro experimental se puede observar que al comienzo de los ciclos de descarga, una deformación en dirección contraria a la dirección de carga aumenta el desplazamiento levemente después de alcanzar el máximo,

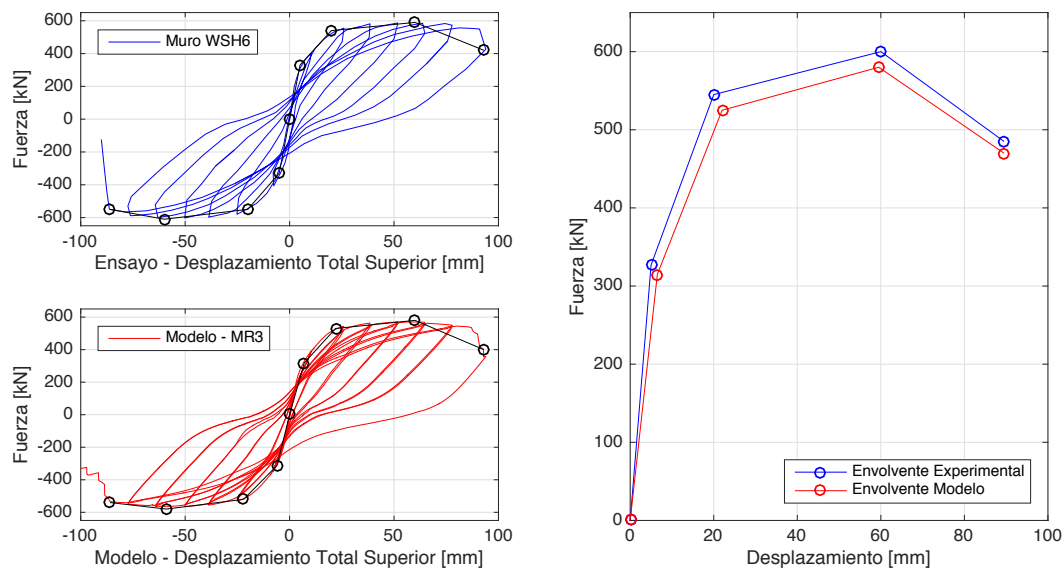


Figura 3.28: Izq: Envolvente total en muro WSH6 y modelo MR3. Der: Promedio de envolvente en muro WSH6 y modelo MR3

Tabla 3.17: Parámetros de comparación - WSH6 vs MR3

	Puntos en la Envolvente							
	Agrietamiento		Fluencia		$Corte_{max}$		$Desplazamiento_{max}$	
	$D_{tot,cr}$	V_{cr}	$D_{tot,Y}$	V_Y	$D_{tot,U}$	V_U	$D_{tot,X}$	V_X
Unidades	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]
Test	5.00	327	20.00	545	60.00	600	89.50	485
Modelo	6.50	315	22.00	525	59.50	580	89.50	470
Modelo/Test	1.30	0.96	1.10	0.96	0.99	0.97	1.00	0.97

Tabla 3.18: Comparación de Rigideces - WSH6 vs MR3

	Rigidez	
	Inicial	Secante en Fluencia
	$K_{ini,tot}$	$K_{Y,tot}$
Unidades	$[kN/mm]$	$[kN/mm]$
Test	65	27
Modelo	49	24
Modelo/Test	0.74	0.88

Tensión y Deformación

En las figuras 3.29 y 3.30 se señala la distribución de esfuerzos (S) y deformaciones (E) para un drift de 1 % y para d_{max} .

En los esfuerzos podemos ver que ambos gráficos son similares, gran parte debido a que al 1 % es muy similar la resistencia del muro que para d_{max} , como lo muestra el segundo y tercer punto de la envolvente en la figura 3.28. Los esfuerzos son lógicos al desplazamiento positivo, con altas tensiones en la base, sobretodo en los extremos confinados de dirección contraria en S11 y S22 entre extremos, y en la zona superior donde está siendo cargado. Los esfuerzos de corte S12 se distribuyen superiormente en el muro, siendo máximos en el extremo derecho inferior comprimido.

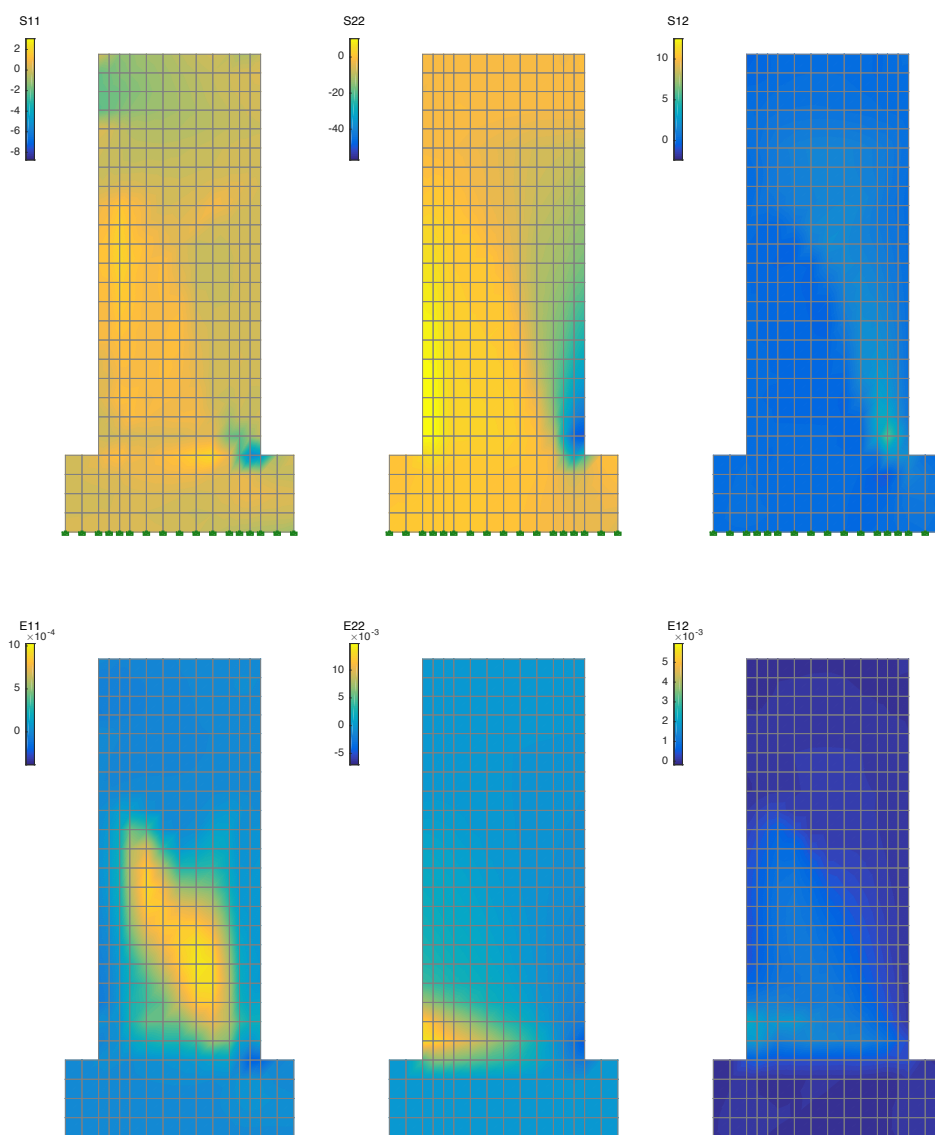


Figura 3.29: Tensión y deformación modelo MR3 al 1 % de deriva

En las deformaciones si se presentan cambios entre el 1% y d_{max} , presentando deformaciones axiales más concentradas en la zona inferior para E11 y E22. En el caso de E11 disminuye la altura de las concentraciones de deformación, probablemente inicialmente debido a la carga vertical aplicada, y luego, prevalece la deformación por el desplazamiento superior, concentrándose en la zona inferior. Para E22 aumentan las deformaciones debido a mayor cantidad de ciclos y desplazamientos, sobretodo en las zonas a compresión y tracción ubicadas en los extremos inferiores. Finalmente, la deformación tangencial E12 aumenta en la zona inferior, zona de principal deformación debido a los ciclos.

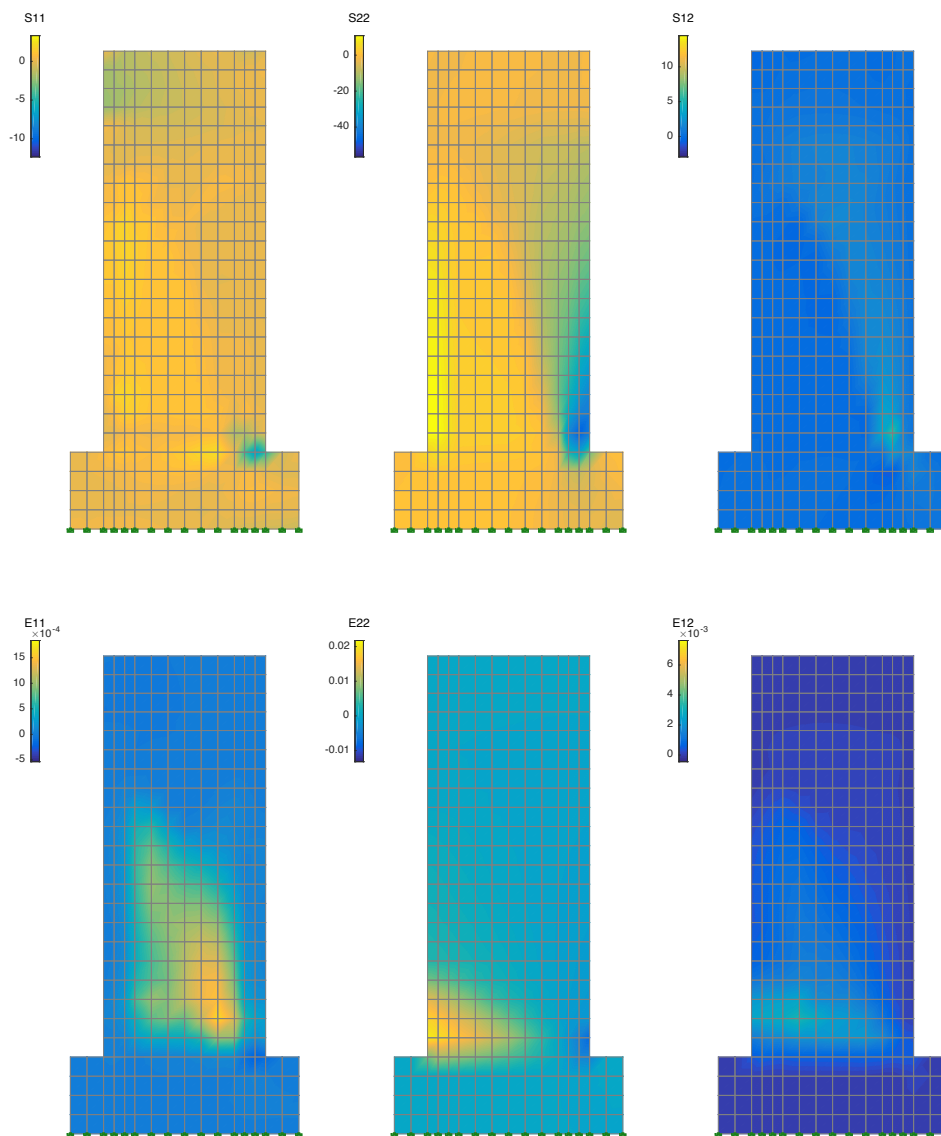


Figura 3.30: Tensión y deformación modelo MR3 al d_{max} - 2.22% de deriva

3.4. Caso 4 - MR4

3.4.1. Investigación: Hysteretic Behavior of Reinforced Concrete Structural Walls

Vallenas, J., Bertero, V. y Popov, E., (1979) [15] estudiaron el comportamiento histórico de muros de hormigón armado.

El objetivo principal de esta investigación es ver el comportamiento histórico de los muros, analizando el aporte de los estribos de confinamiento, el comportamiento de muros rectangulares con columnas en sus elementos de borde, distintas razones de momento-corte, ciclos de carga monótonos y cíclicos y procedimientos de reparación de muros.

Para el comportamiento histórico se estudia la rigidez, ductilidad, contribución a la deformación de los diferentes elementos, capacidad de disipación de energía y modos de falla.

El experimento consistía en ensayar cuatro muros de hormigón armado de tres pisos, los cuales representaban los muros de corte de la parte baja de dos edificios de 7 y 10 pisos. Posterior a realizar los ensayos y lograr falla en estos muros, estos se reparaban y se volvían a ensayar. Además, se realizaron modelos y análisis de los muros estudiados, con el fin de comparar y lograr representar de forma correcta el comportamiento de los muros.

Los especímenes fueron cargados lateralmente, dos gatos hidráulicos en respectivos pisos y uno en el extremo superior que tenía la mayor capacidad, además de otros dos gatos hidráulicos en la zona superior para ejercer esfuerzo axial a los muros. La mitad de los muros se ensayaron con un ciclo monótono y la otra mitad cíclico. Posterior a su reparación, se ensayo el tipo de ciclo no realizado en la primera vez a cada muro.

Como conclusión de esta investigación se asevera que las normas utilizadas subestiman la capacidad de los muros diseñados, tanto para resistir esfuerzos a flexión como esfuerzos de corte. A su vez, los materiales logran un mayor desempeño que el de diseño. También se asevera el correcto uso estimando la rigidez de los muros por medio de la teoría de vigas, la estabilidad ante cargas monótonas y cíclicas, deformación general del muro y la disipación y absorción de energía.

Con respecto a las técnicas de reparación utilizadas, se menciona que tuvieron un correcto funcionamiento, aunque claramente no se logró obtener los mismos ni mejores resultados que un muro nuevo. También menciona recomendaciones para futuras investigaciones en cuanto a modelos y experimentos.

Se utiliza el muro S6 para modelar y comparar su respuesta en esta investigación, modelo asignado con el nombre de MR4.

3.4.2. Especimen de Estudio: S6

El muro S6 consiste en un muro de hormigón armado, con tres losas transversales en distintas alturas, carga vertical estática en los extremos superiores, y cargas aplicadas horizontalmente en su viga de carga superior y en dos puntos inferiores, en las losas del muro. En la figura 3.31 se muestra la disposición del ensayo. La carga horizontal impuesta es de forma cíclica. Cabe destacar que el muro S6 fue ensayado de forma horizontal, por lo que el efecto del peso propio afecta solo transversal al plano del muro.

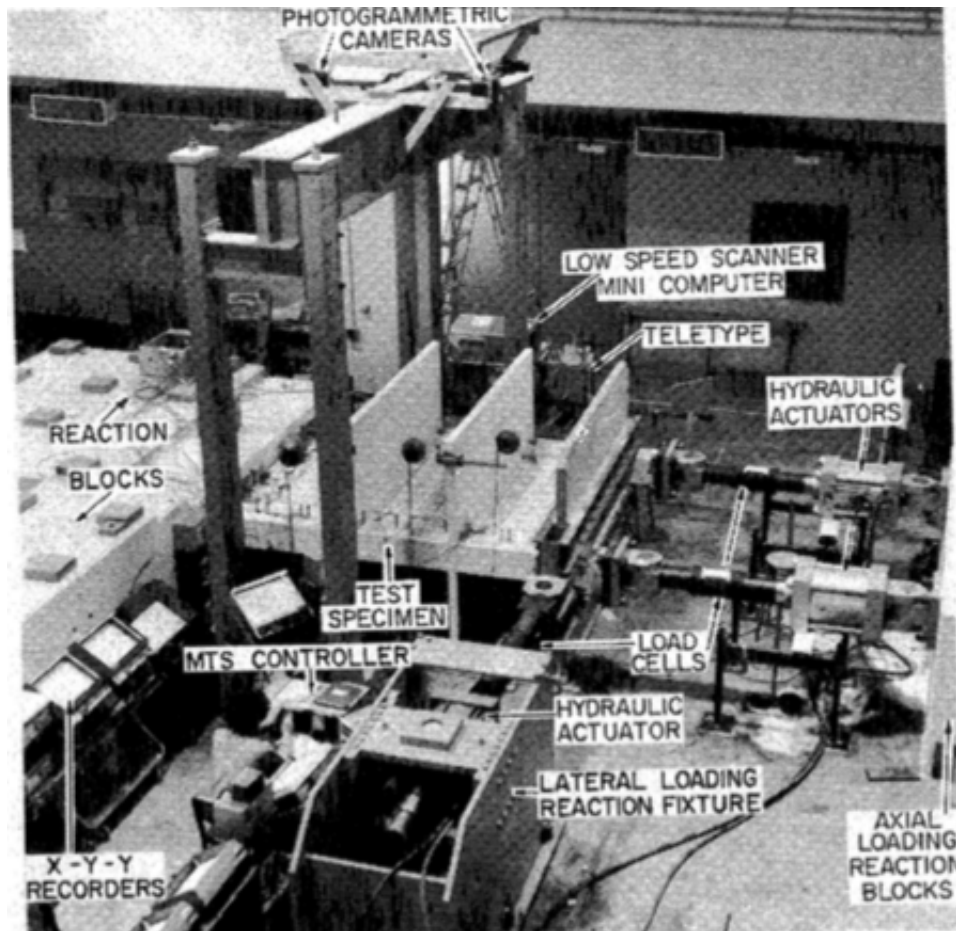


Figura 3.31: Ensayo realizado en el muro S6. (Imagen: Vallenas, 1979 [15])

Dimensiones y Geometría

La altura, ancho y espesor del muro se especifican en la tabla 3.19. El muro S6 presenta vigas transversales en tres puntos de su altura, las cuales se pueden apreciar en la figura 3.32, incluyendo sus dimensiones y vista en dos planos.

Tabla 3.19: Geometría Muro S6

Dimensión	Valor
$H_w [mm]$	3048
$L_w [mm]$	2412
L_{conf}	267
$T_w [mm]$	114
H_w/L_w	1.26
L_w/T_w	21.16
s/d	2.14

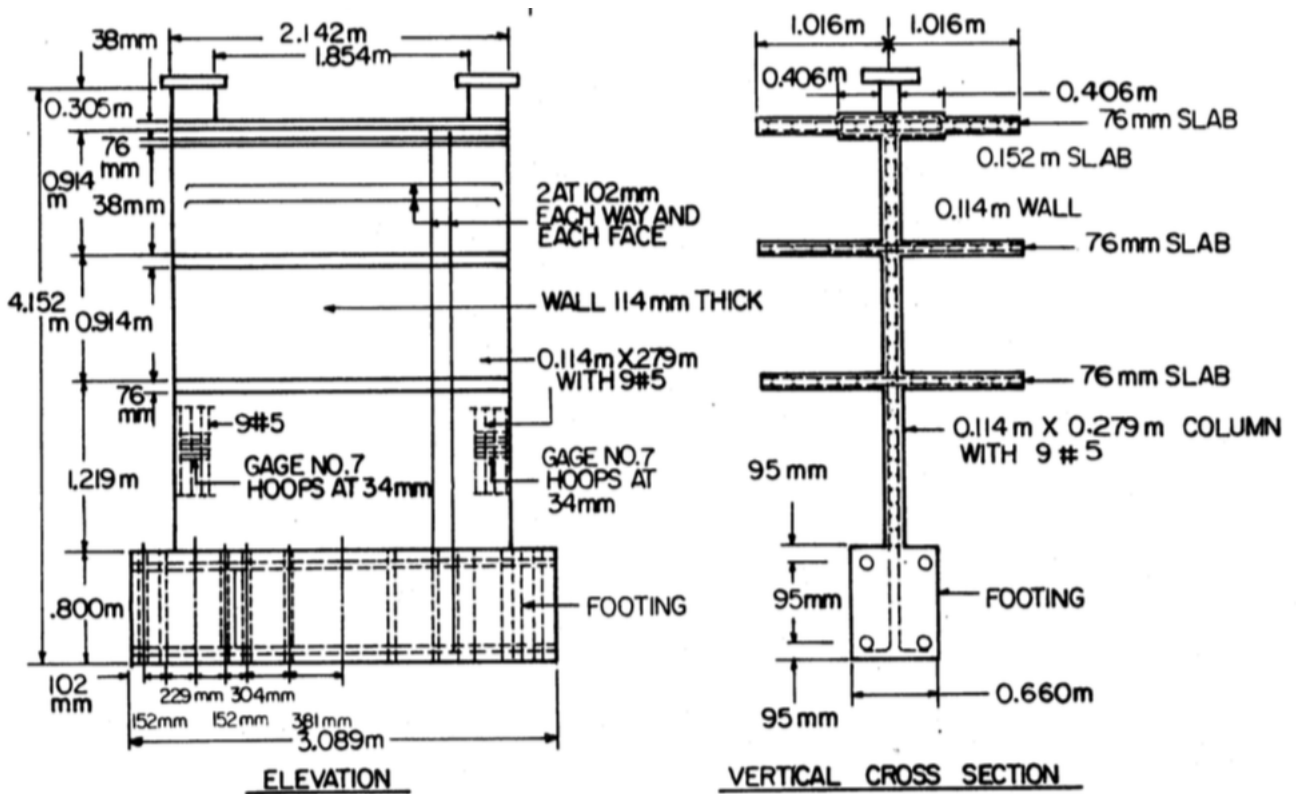


Figura 3.32: Dimensiones y geometría del Muro S6

Armadura de refuerzo

En la tabla 3.20 se presentan las cuantías según sección del muro. En la figura 3.33 se puede ver la sección transversal de armadura del muro.

Tabla 3.20: Cuantías de refuerzo en Muro S6

Cuantía	[%]
ρ_v	0.56
ρ_h	0.54
$\rho_{estribos}$	0.84
ρ_{conf}	5.86

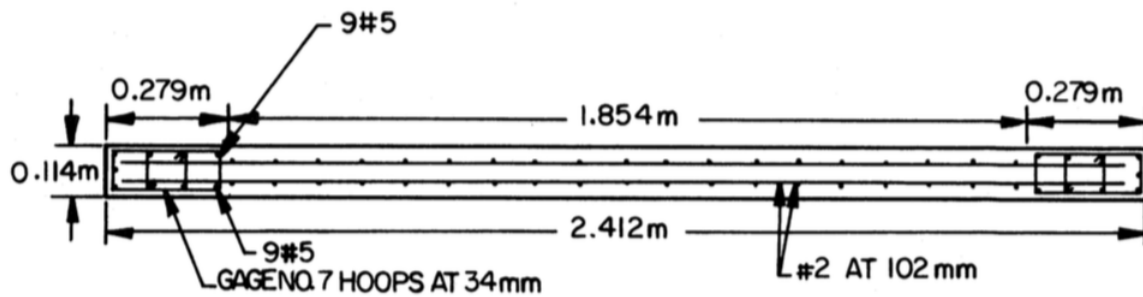


Figura 3.33: Refuerzo de armadura utilizado en Muro S6. (Imagen: Vallenas, 1979 [15])

Materiales

Para cada uno de los materiales, tanto barras de acero como hormigón, se especifican las capacidades de resistencia y módulo de Young utilizados para el diseño y ensayados para el muro S6, en la tabla 3.21.

Tabla 3.21: Características de materiales

Característica	Diseño	Experimental
f'_c [MPa]	28	35
f_{yv} [MPa]	414	507
f_{yh} [MPa]	414	507
f_{yconf} [MPa]	414	482
f_{ys} [MPa]	414	440
$E_{sConfinamiento}$ [MPa]	200,000	216,000
$E_{sEstribos}$ [MPa]	200,000	211,000
$E_{sCentrales}$ [MPa]	200,000	190,000

Cargas aplicadas

Las cargas aplicadas en este muro consisten en cargas aplicadas de forma horizontal en la viga de carga superior, de forma cíclica, con un protocolo de carga mostrado en la figura 3.34. A su vez, dos cargas aplicadas en dos puntos inferiores, de menor capacidad a la ejercida en la zona superior. En sus extremos superiores, cargas verticales estáticas de 299 [kN] aplicadas

en cada extremo. El muro no tenía influencia de su peso propio debido a que el ensayo fue realizado con el muro dispuesto de forma horizontal.

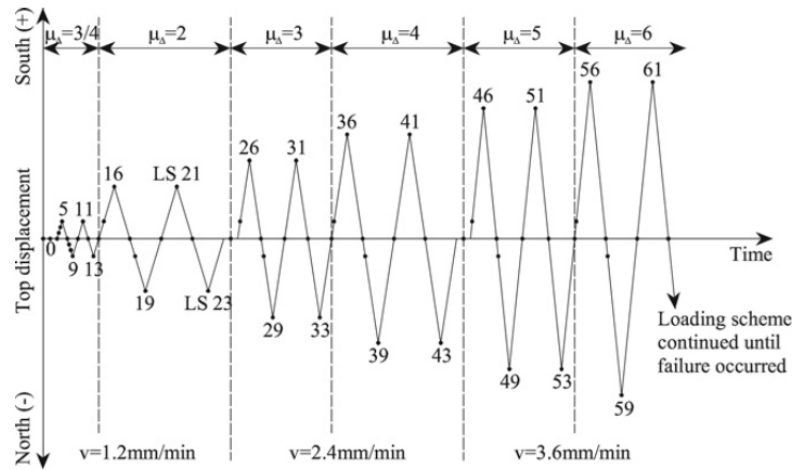


Figura 3.34: Protocolo de desplazamiento de ensayo muro S6. (Imagen: Vallenas, 1979 [15])

3.4.3. Modelo

Mallado

El muro se modeló por medio de catorce elementos horizontales, dos elementos a cada extremo representa la zona confinada, de ancho 130 y 137 [mm] en cada costado, y otros diez elementos centrales de ancho 188 [mm]. La partición vertical consta de 17 elementos de una altura de 184 [mm] cada uno. El pedestal de fundación está constituido por cuatro elementos verticales de altura 184 [mm] y 18 elementos horizontales de ancho similar al dispuesto en los elementos de muro adyacentes. La viga de carga consta de 4 elementos verticales dispuestos sobre el muro, con igual cantidad de elementos horizontales que el muro. En la figura 3.35 se puede ver el mallado de forma gráfica.

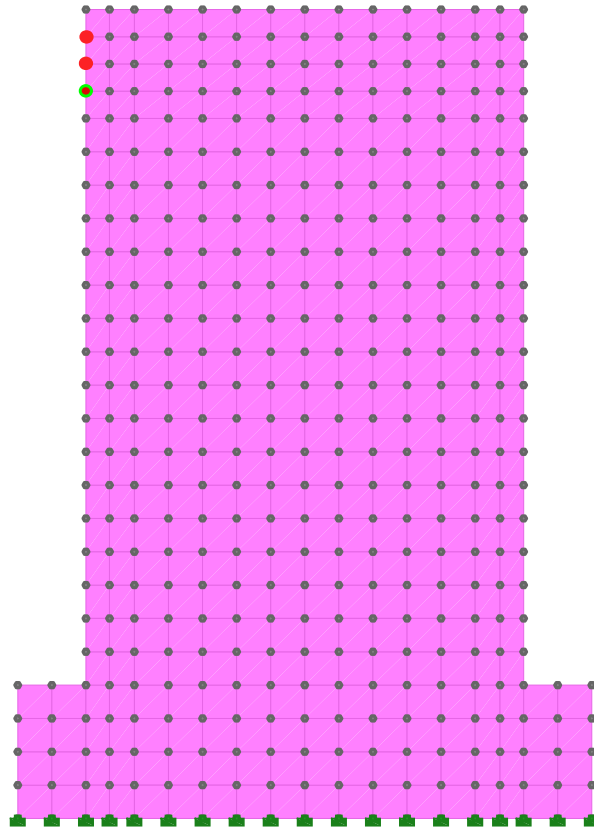


Figura 3.35: Partición Muro MR4

Cargas

Las fuerzas verticales actuantes sobre el modelo se ejercieron en 3 nodos de cada extremo superior para asimilar las cargas aplicadas en las columnas a los extremos del muro, las cuales correspondían a 299 kN distribuidos. Las cargas horizontales fueron aplicadas en los tres nodos izquierdos superiores señalados en rojo en la figura 3.35. En verde, el nodo de registro de desplazamientos.

Coefficientes del modelo

Los coeficientes de daño utilizados se señalan en la tabla 3.22.

Tabla 3.22: Coeficientes de modelo - Modelo MR4

Coeficiente	Valor
α_{1conf}	0.1
α_{2conf}	0.1
α_1	0.1
α_2	0.1
K_{conf}	1.08

3.4.4. Análisis y Resultados

El muro S6 presentaba la complejidad de tener losas transversales, con anchos considerables pero de baja altura. Además, fue ensayado con el muro dispuesto horizontalmente, a diferencia de los demás muros en este trabajo. Es un muro con una razón de aspecto relativamente baja (1.26), por lo que, se espera un comportamiento dominado por el corte, con altas tensiones de cizalle. A partir de su reporte, se verifica que en los últimos ciclos presentó separación entre el muro y la fundación, además de su falla por pandeo en sus columnas a causa de aplastamiento.

Resultado Experimental vs Modelo

El muro S6 presenta el desafío de tener que modelar las losas transversales en su sección, las cuales fueron modeladas como vigas elásticas nodo a nodo en las alturas en que estaban presentes en el muro experimental. Cuando se considero daño con coeficientes como los demás modelos, el muro no logro converger, por lo tanto se redujo al mínimo de 0.1, y se obtuvieron resultados acordes al ensayo. Esto fue probablemente debido a la deformación excedida del modelo, lo que dificultaba la capacidad de converger en los nodos donde se unían las vigas elásticas con el muro.

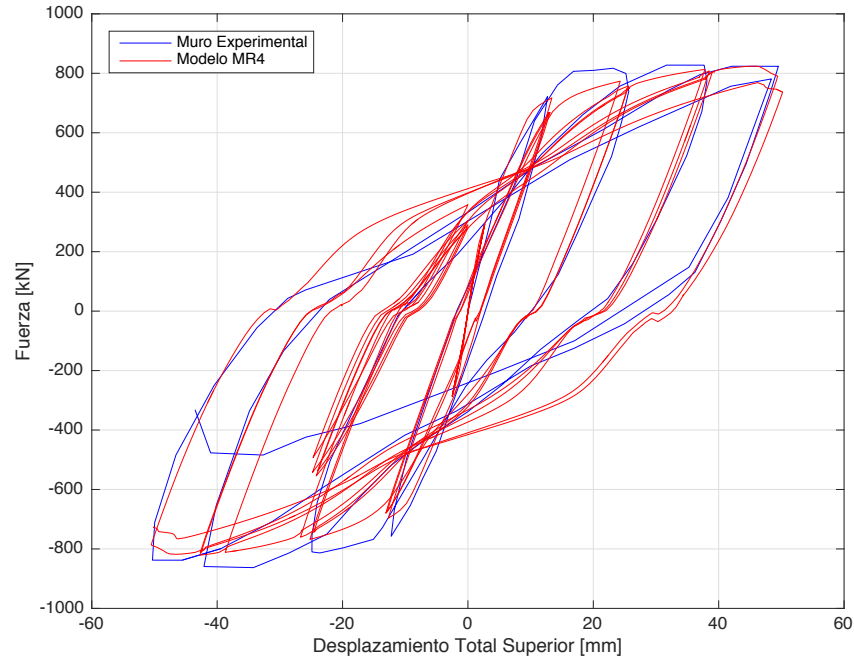


Figura 3.36: Gráfico de fuerza vs deformación total superior para muro experimental S6 y modelo MR4

La figura 3.36 permite ver ciclos con igualdad de resistencia para el modelo como el muro experimental, logrando representar el comportamiento del muro. Además, se presentan similares curvas a las del modelo MR2 para cuando la fuerza es igual a 0, además de tener un último ciclo con mayor resistencia, degradándose menos que el muro experimental y fallando justo en los pasos posteriores a la falla en -40 [mm] del muro experimental.

Comparación de Envolvente

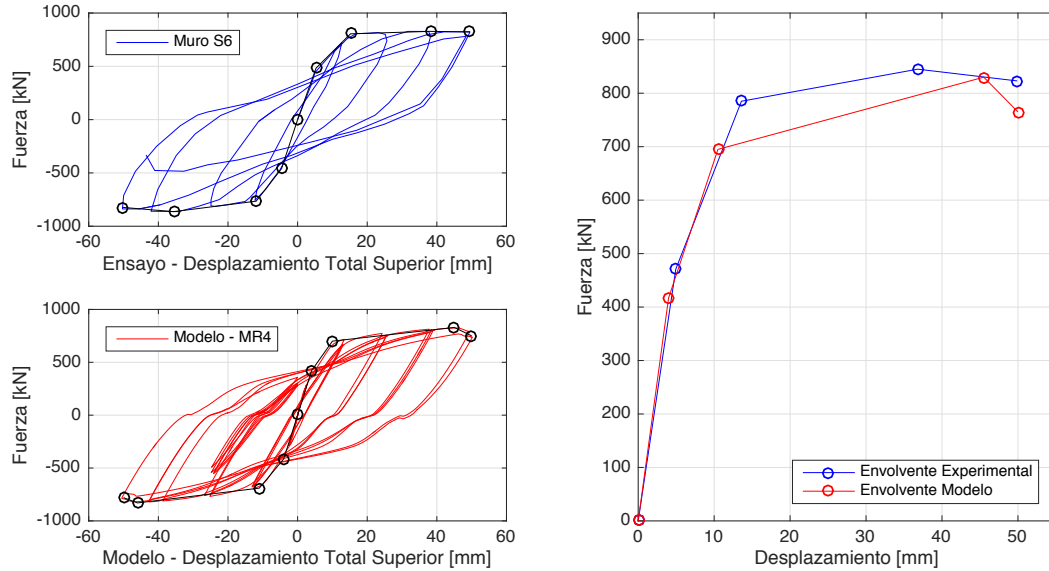


Figura 3.37: Izq: Envolvente total en muro S6 y modelo MR4. Der: Promedio de envolvente en muro S6 y modelo MR4

A partir de los puntos de la envolvente de la figura 3.37 a la izquierda, se puede comparar una mayor rigidez del muro experimental que el modelo inicialmente, logrando un 9% más de pendiente. En el segundo punto, correspondiente a la fluencia, es el modelo el que adquiere mayor pendiente, un 8% más, logrando similares valores de resistencia máxima y de deformación final que el muro experimental.

Al promediar los puntos positivos y negativos (figura 3.37 a la derecha), podemos ver un comportamiento de iguales características iniciales, pero que después se diferencian en la fluencia y el tercer punto debido a que el muro experimental alcanza el máximo antes y lo mantiene, a diferencia del modelo MR4 que lo alcanza cercano a los 40 [mm] de desplazamiento. También en el punto de máximo desplazamiento difiere entre el modelo y el experimental, disminuido producto del degradamiento del último ciclo.

Tabla 3.23: Parámetros de comparación - S6 vs MR4

	Puntos en la Envolvente							
	Agrietamiento		Fluencia		$Corte_{max}$		$Desplazamiento_{max}$	
	$D_{tot,cr}$	V_{cr}	$D_{tot,Y}$	V_Y	$D_{tot,U}$	V_U	$D_{tot,X}$	V_X
Unidades	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]
Test	4.95	471	13.59	785	36.83	845	49.91	822.9
Modelo	4.00	417	10.50	695	45.50	830	50.00	792.5
Modelo/Test	0.81	0.89	0.77	0.89	1.24	0.98	1.00	0.96

Tabla 3.24: Comparación de Rigideces - S6 vs MR4

	Rigidez	
	Inicial	Secante en Fluencia
	$K_{ini,tot}$	$K_{Y,tot}$
Unidades	$[kN/mm]$	$[kN/mm]$
Test	95	58
Modelo	104	54
Modelo/Test	1.09	0.92

Tensión y Deformación

En las figuras 3.38 y 3.39 se señala la distribución de esfuerzos (S) y deformaciones (E) para un drift de 1% y para d_{max} .

En los esfuerzos podemos ver que el modelo, para ambos estados, se comportan de similar forma, con altas tensiones en los extremos inferiores, y con signos correspondientes debido al desplazamiento positivo. Para el caso del cizalle, S12, se obtiene un aumento leve en el caso del d_{max} con respecto al 1%, concentrados en la zona de la base en el extremo izquierdo traccionado.

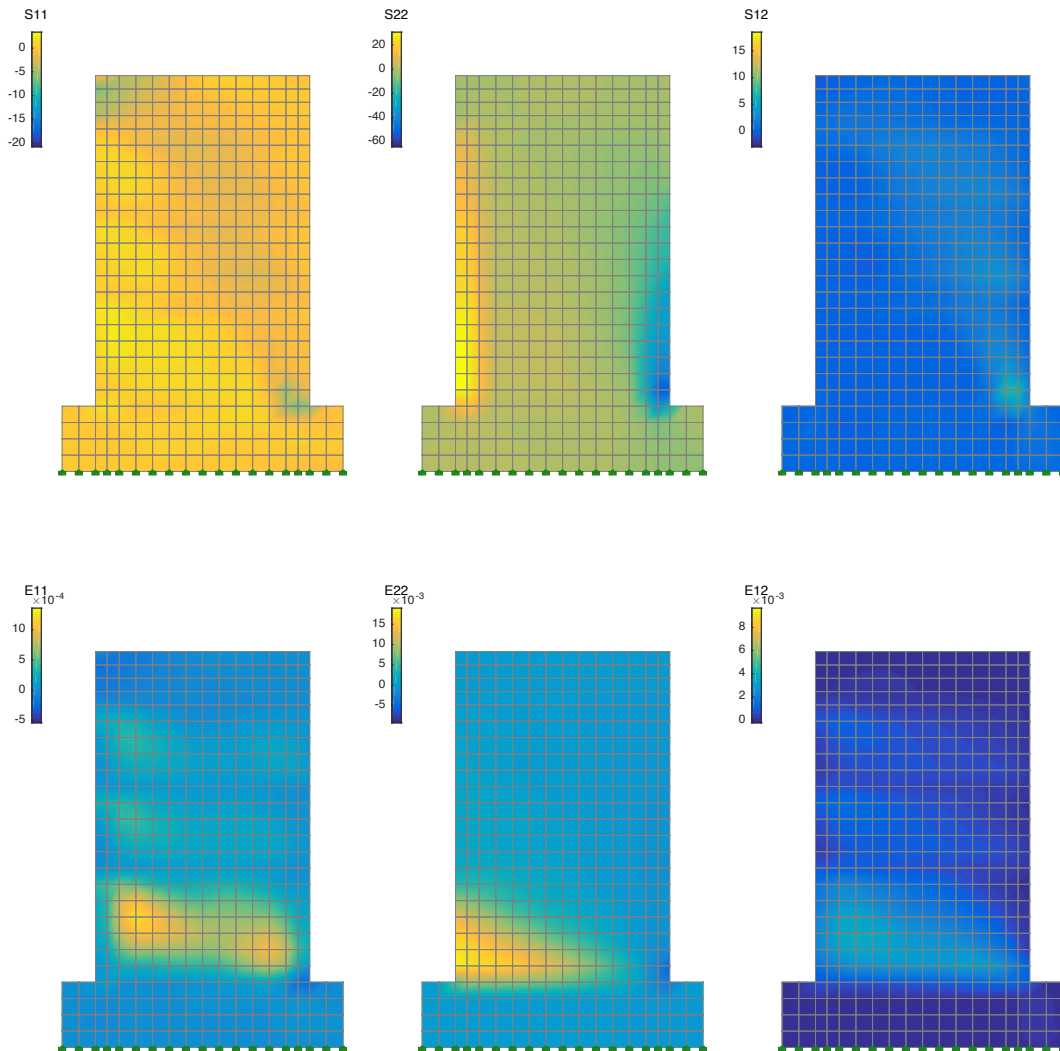


Figura 3.38: Tensión y deformación modelo MR4 al 1% de deriva

Los cambios en las deformaciones se presentan principalmente para E11, donde se concentran en la zona inferior. En el caso de la deformación tangencial, E12, aumentan las

deformaciones cercanas a la base pero disminuyen aquellas distribuidas a lo largo del muro. En E22 se mantiene similar entre el 1% y d_{max} , manteniendo el sentido lógico de deformación dada la deformación positiva superior.

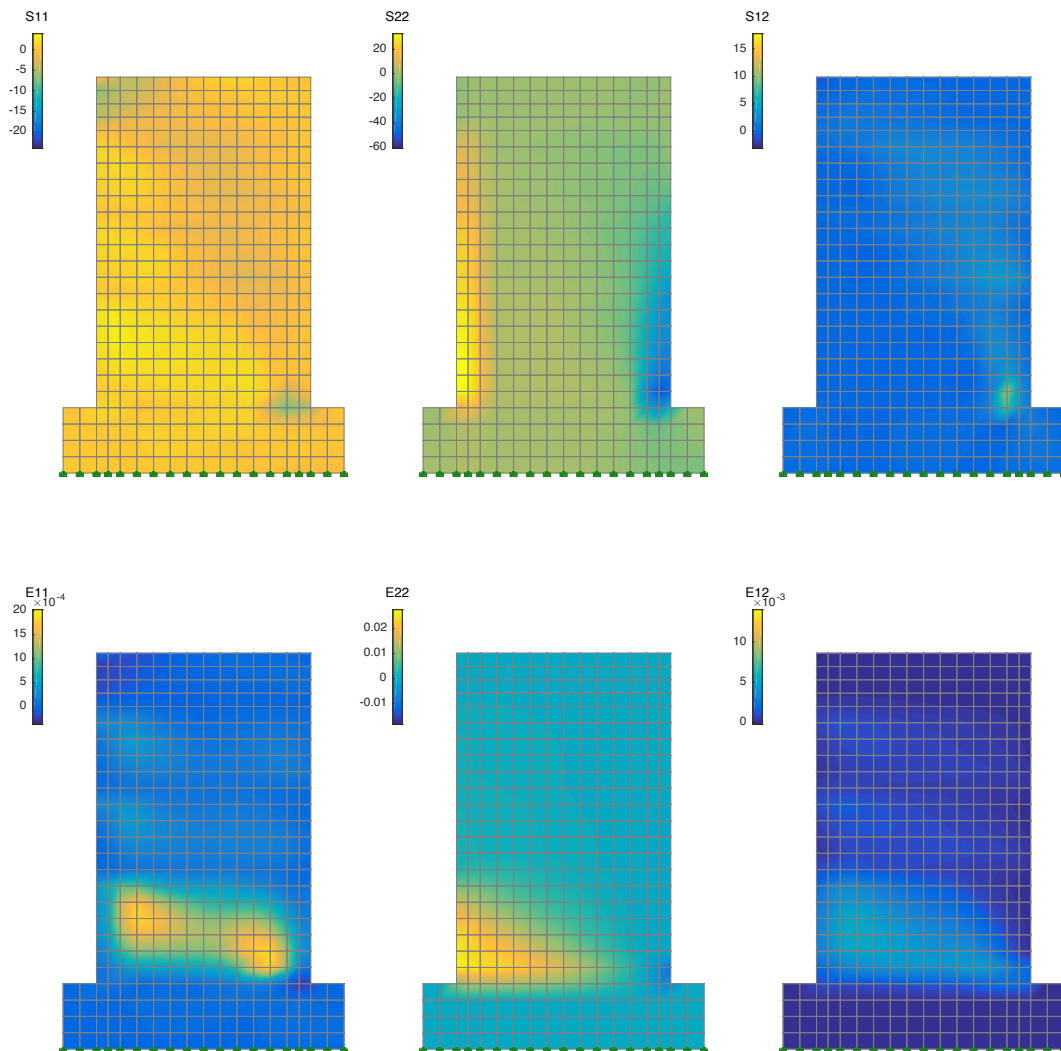


Figura 3.39: Tensión y deformación modelo MR4 al d_{max} - 1.64% de deriva

3.5. Caso 5 - MR5

3.5.1. Investigación: Earthquake Resistant Structural Walls - Test of Isolated Walls

El presente reporte corresponde a un estudio realizado por los investigadores Oesterle, R., Fiorato, A., Johal, L., Carpenter, J., Russell, H. y Corley, W., en el *Structural Development Section of the Portland Cement Association*, con respecto a la resistencia de muros estructurales ante terremotos.

Los objetivos incluían determinar las características de carga y deformación de los muros, en función de su composición. A su vez, determinar el nivel de ductilidad y energía de disipación, analizando el nivel requerido según el muro. Verificar la resistencia a la flexión y al corte de los muros para cargas cíclicas, y lograr comparar ésta con cargas monótonas. Finalmente, proponer procedimientos de diseño para asegurar una adecuada capacidad de resistencia y disipación de energía.

El programa experimental de la investigación incluyó la aplicación de cargas cíclicas sobre distintas configuraciones de muros aislados, los cuales eran representaciones de muros completos escalados en 1/3. El estudio contempla 9 muros, donde las variaciones entre ellos eran: forma de la zona transversal (rectangular, con columnas en los extremos y doble "T"), cantidad de refuerzo vertical dispuesto, además del nivel de estribos y confinamiento.

Los especímenes fueron cargados en su zona superior, con incrementos de desplazamiento de 25 mm, después de 3 ciclos completos. Se realizaron ensayos de vibraciones libres al alcanzar determinadas cargas en el muro, a modo de calcular el periodo y amortiguamiento.

Todos los especímenes lograron una mayor capacidad, tanto a flexión como a corte, del diseño A.C.I., sin embargo, se observó una capacidad menor a flexión al utilizar las propiedades reales de los materiales y no las de diseño. Todos los muros mostraron una alta capacidad de deformación inelástica, a pesar de que algunos no fueron diseñados con ese propósito. Sobre 1.83[m] de altura de los muros se vio reiteradamente la fluencia del refuerzo tanto vertical como horizontal, aunque el daño se presentó normalmente en la zona baja. Los muros mostraron dos comportamientos dependiendo de si el esfuerzo de corte era alto o bajo, mostrando diferencias claras en su respuesta.

Se concluyeron las siguientes aseveraciones:

- El diseño de acuerdo al A.C.I. Building Code de 1971, sin ocupar diseño sísmico, logra las capacidades ACI para flexión y corte. Además, cumple con una alta capacidad de deformación inelástica para cargas cíclicas.
- Existe un alto aporte a la capacidad inelástica por medio del confinamiento en los elementos de borde. El uso de estribos y confinamiento ayuda a mantener íntegro los bordes de muro y su centro de hormigón, aumentando su rigidez, reduciendo el daño, facilitando la reparación, aumenta la resistencia al corte. El refuerzo vertical de confinamiento permite alcanzar altas tensiones en los bordes, siendo diseñado bajo los criterios

de la ACI 1971.

- Se muestra en detalle ambos tipos de comportamientos, determinados por los esfuerzos de corte dependiendo de si v_{max} era mayor o menor a $3.0\sqrt{f'_c}$
- Los desplazamientos debidos al corte son la mayoría del desplazamiento total de muros ensayados cíclicamente.

De esta investigación se utiliza el muro R2 para modelar y comparar los resultados obtenidos, modelo asignado el nombre de MR5.

3.5.2. Muro de Estudio: R2

El muro R2 es un muro de hormigón armado rectangular, apoyado sobre un pedestal y sometido a una carga superior horizontal de forma cíclica, controlado por el desplazamiento superior posterior al agrietamiento inicial. En la figura 3.40 se muestra el montaje experimental.

Es un muro particular dado que durante su ensayo se notaron graves efectos de pandeo global en el muro durante los primeros ciclos, lo cual fue arreglado al décimo ciclo (desplazamiento ~ 12 [mm]) agregando un soporte lateral para evitar su falla completa por pandeo.

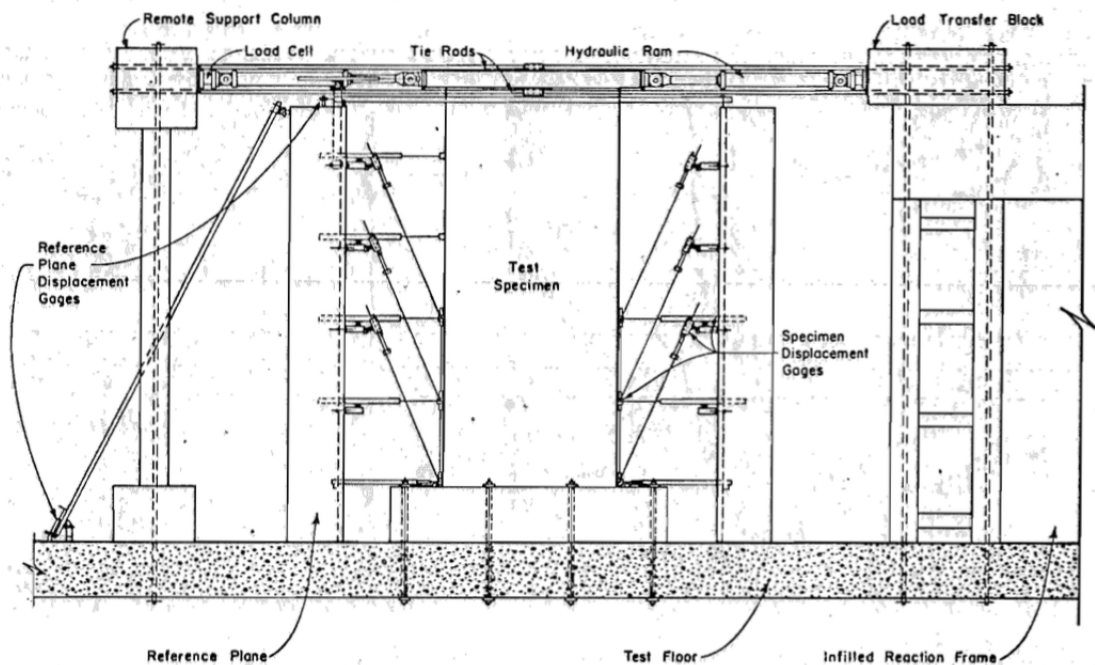


Figura 3.40: Ensayo experimental para Muro R2. (Imagen: Oesterle et al., 1980 [8]).

Dimensiones y Geometría

La altura, ancho y espesor del muro se especifican en la tabla 3.25. En la figura 3.41 se puede ver el muro en proyección, con sus respectivas medidas.

Tabla 3.25: Geometría Muro R2

Dimensión	Valor
$H_w [mm]$	4470
$L_w [mm]$	1905
$L_{conf} [mm]$	190
$T_w [mm]$	102
H_w/L_w	2.35
L_w/T_w	18.75
s/d	2.67

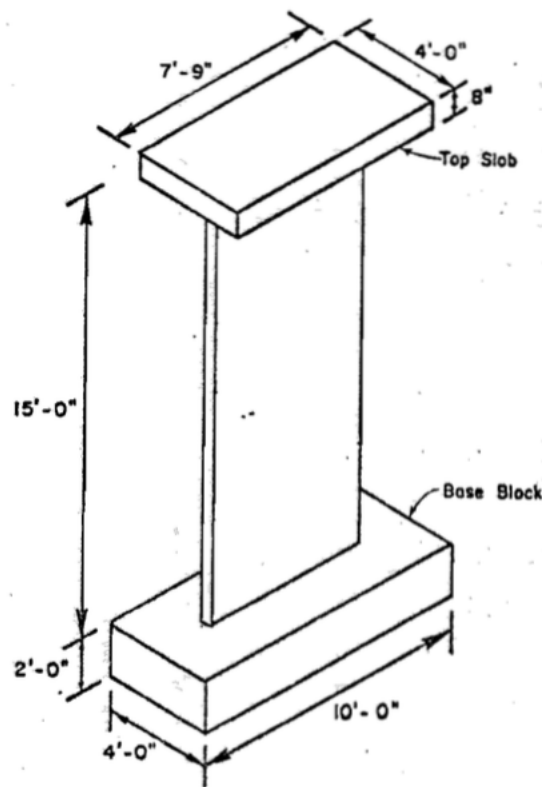


Figura 3.41: Muro R2 en vista 3D. (Imagen: Oesterle et al., 1980 [8])

Cuantía de refuerzo

En la tabla 3.26 se muestran las cuantías del muro experimental R2. La disposición de los refuerzos se muestran en la figura 3.42, en donde todas las unidades están en pulgadas.

Tabla 3.26: Cuantías de refuerzo en Muro R2

Cuantía	[%]
ρ_v	0.25
ρ_h	0.31
$\rho_{estribos}$	1.84
ρ_{conf}	3.93

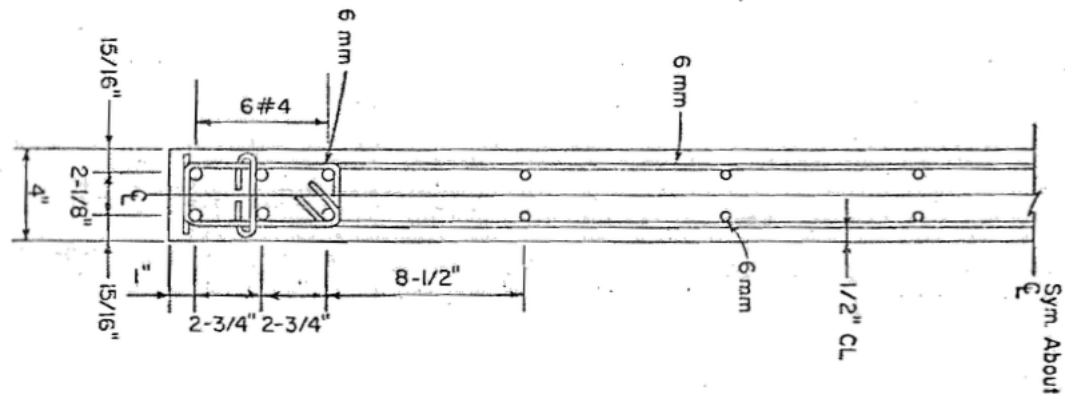


Figura 3.42: Armadura del muro R2. (Imagen: Oesterle et al., 1980 [8])

Materiales

Para cada uno de los materiales, tanto barras de acero como hormigón, se especifican las capacidades de resistencia y módulo de Young utilizados en el diseño y determinado en los ensayos, en la tabla 3.27.

Tabla 3.27: Características de materiales

Característica	Diseño	Experimental
f'_c [MPa]	41.1	46.4
f_{yv} [MPa]	414	535
f_{yh} [MPa]	414	535
f_{yconf} [MPa]	414	455
f_{ys} [MPa]	414	535
E_s [MPa]	200,000	224,777

Cargas aplicadas

La carga fue aplicada en la zona superior del muro, compuesta de hormigón armado de mayor espesor y mayor ancho que el muro, lo cual permitía aplicar las cargas sin sufrir grandes

deformaciones y lograr un correcto traspaso de la carga hacia el muro. La carga horizontal se aplicó de forma cíclica, realizando tres ciclos antes de aumentar la amplitud de carga. En la figura 3.43 se puede ver el protocolo de carga de este ensayo. Cabe recalcar que este muro sufrió graves efectos de pandeo, por lo que al décimo ciclo de carga se escogió añadir soportes laterales en el primer piso.

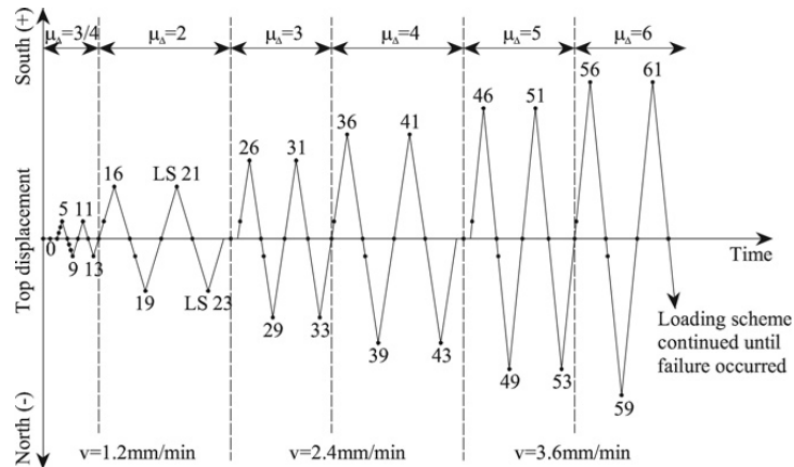


Figura 3.43: Protocolo de desplazamiento de ensayo muro R2. (Imagen: Oesterle et al., 1980 [8])

3.5.3. Modelo

Mallado

El modelo del muro MR5 contempla dos elementos laterales de confinamiento, uno de 90 y 100 [mm] a cada costado de ancho, y nueve elementos centrales de ancho 170 [mm]. La disposición vertical corresponde a 27 elementos de altura 166 [mm]. El muro se sustenta centrado sobre un pedestal de fundación con 17 elementos horizontales y 4 elementos verticales, en donde el ancho dependiente los elementos de muro superiores y la altura es 163 [mm]. Su viga de carga en la zona superior, replica la sección utilizada en su pedestal, solo que se compone de 15 elementos horizontales. El modelo se puede ver de forma gráfica en la figura 3.44.

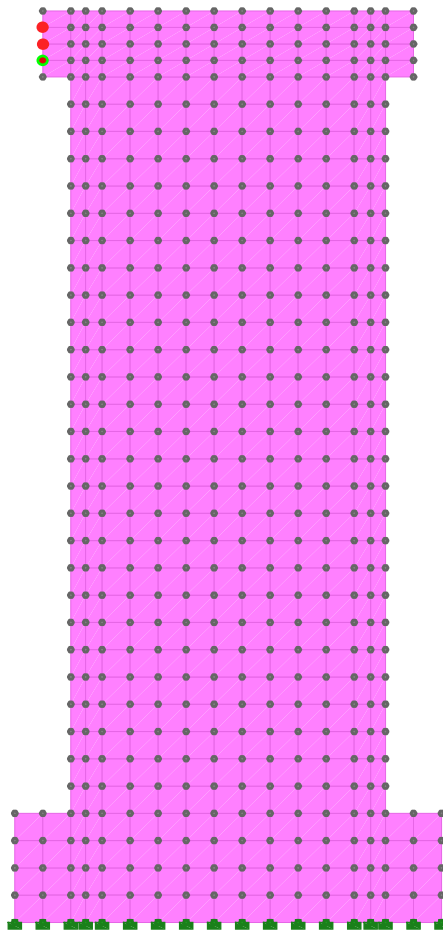


Figura 3.44: Partición Muro MR5

Cargas

El muro fue cargado verticalmente en los 11 nodos centrales superiores, aplicando una carga de -15 [kN] distribuidos, equivalentes al peso del soporte superior. La carga horizontal cíclica fue aplicada en tres nodos izquierdos laterales señalados en rojo en la figura 3.44. En verde, el nodo de registro de desplazamientos.

Coefficientes de daño

Los coeficientes utilizados en el programa se señalan en la tabla 3.28.

Tabla 3.28: Coeficientes de Daño - Modelo MR5

Coeficiente	Valor
α_{1conf}	0.15
α_{2conf}	0.4
α_1	0.15
α_2	0.4
K_{conf}	1.16

3.5.4. Análisis y Resultados

El muro R2 es un muro con razón de aspecto de 2.35, y el muro más alto modelado y con más ciclos realizados. En su ensayo experimental, presento problemas de graves efectos de pandeo global, deformando el muro fuera del plano. Cercano al 10 ciclo, cercano a los 12 [mm] de desplazamiento superior, se detuvo el ensayo y se añadieron soportes laterales para continuar el ensayo. Finalmente, en los últimos ciclos, fallo por pandeo local de barras en su confinamiento inferior debido al gran aplastamiento en sus extremos y los desplazamientos fuera del plano por pandeo global.

Resultado Experimental vs Modelo

Este muro presentó grandes dificultades para realizar su modelo, debido a que es un muro alto (4.45 [m]) y ancho (1.9 [m]), siendo el modelo de más elementos entre los muros modelados, además, su protocolo de carga consta de 39 ciclos, lo cual hacia el proceso más lento a la espera de grandes deformaciones, donde los modelos presentan los principales problemas.

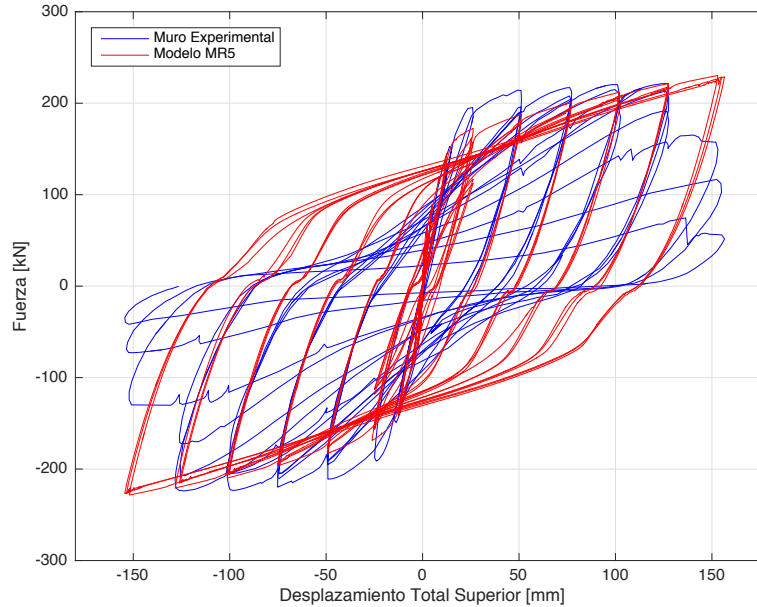


Figura 3.45: Gráfico de fuerza vs deformación total superior para muro experimental R2 y modelo MR5

Los resultados obtenidos se pueden ver en la figura 3.45, donde se obtiene una similar rigidez inicial pero no en fluencia, alcanzando una mayor resistencia en el caso del muro experimental. Posteriormente, el modelo alcanza la misma resistencia que el muro experimental, cercano a un desplazamiento de 125 [mm]. En los ciclos de descarga, aparece una curva para fuerzas cercanas a 0, alejándose en ese punto de la curva experimental a medida que aumenta la carga. En los últimos ciclos, de desplazamientos sobre 150 [mm] el muro experimental sufre un degradamiento sostenido, lo cual el modelo no logra reflejar.

La posible dificultad de reflejar correctamente este modelo se puede deber a los hechos ocurridos en su fase experimental, sumados a que los elementos que modelamos nos permiten ver el movimiento 2D que tenemos en los muros, pero no así el caso de pandeo global donde los desplazamientos ocurren fuera del plano.

Comparación de Envoltente

En el modelo MR5 se ve una resistencia de fluencia menor, tanto en el ciclo de carga como de descarga. Además, el último punto de desplazamiento máximo ocurre en el muro degradado en el ensayo, lo cual nuestro modelo no representa y se mantiene cerca de la fuerza máxima. En el caso de la rigidez inicial, este valor solo es diferente entre el muro y el modelo en un 1 %, además, en la capacidad máxima V_{max} , este valor es reflejado, difiriendo del experimental en un 2 %.

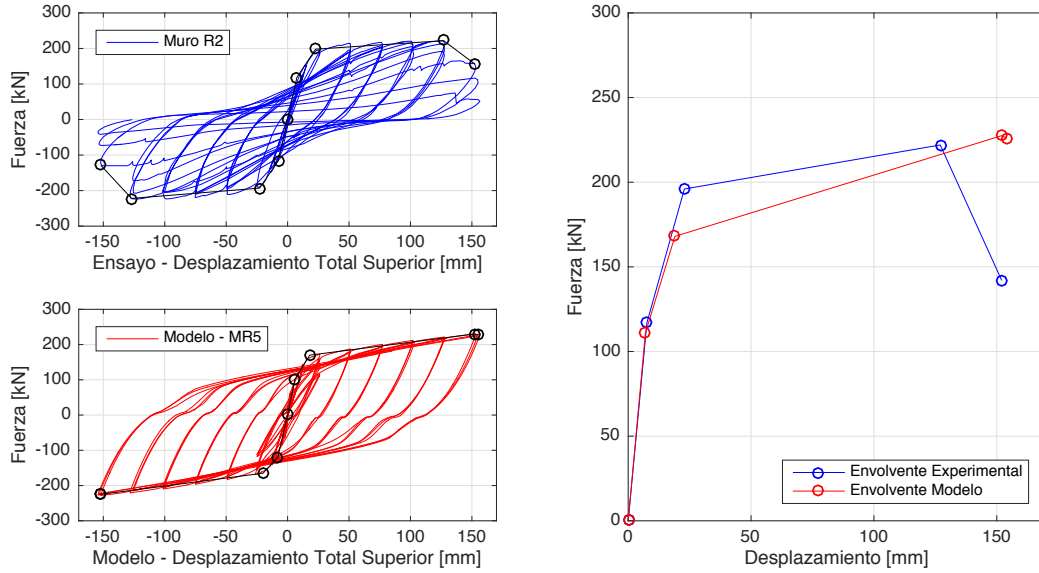


Figura 3.46: Izq: Envolvente total en muro R2 y modelo MR5. Der: Promedio de envolvente en muro R2 y modelo MR5

Tabla 3.29: Parámetros de comparación - R2 vs MR5

	Puntos en la Envolvente							
	Agrietamiento		Fluencia		$Corte_{max}$		$Desplazamiento_{max}$	
	$D_{tot,cr}$	V_{cr}	$D_{tot,Y}$	V_Y	$D_{tot,U}$	V_U	$D_{tot,X}$	V_X
Unidades	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]
Test	7.6	117	23	196	127	222	152	142
Modelo	7.0	111	19	168	152	227	154	226
Modelo/Test	0.92	0.95	0.83	0.86	1.20	1.02	1.01	1.59

Tabla 3.30: Comparación de Rigideces - R2 vs MR5

	Rigidez	
	Inicial	Secante en Fluencia
	$K_{ini,tot}$	$K_{Y,tot}$
Unidades	[kN/mm]	[kN/mm]
Test	15.4	8.56
Modelo	16.0	8.85
Modelo/Test	1.04	1.03

Tensión y Deformación

En las figuras 3.47 y 3.48 se señala la distribución de esfuerzos (S) y deformaciones (E) para un drift de 1% y para d_{max} .

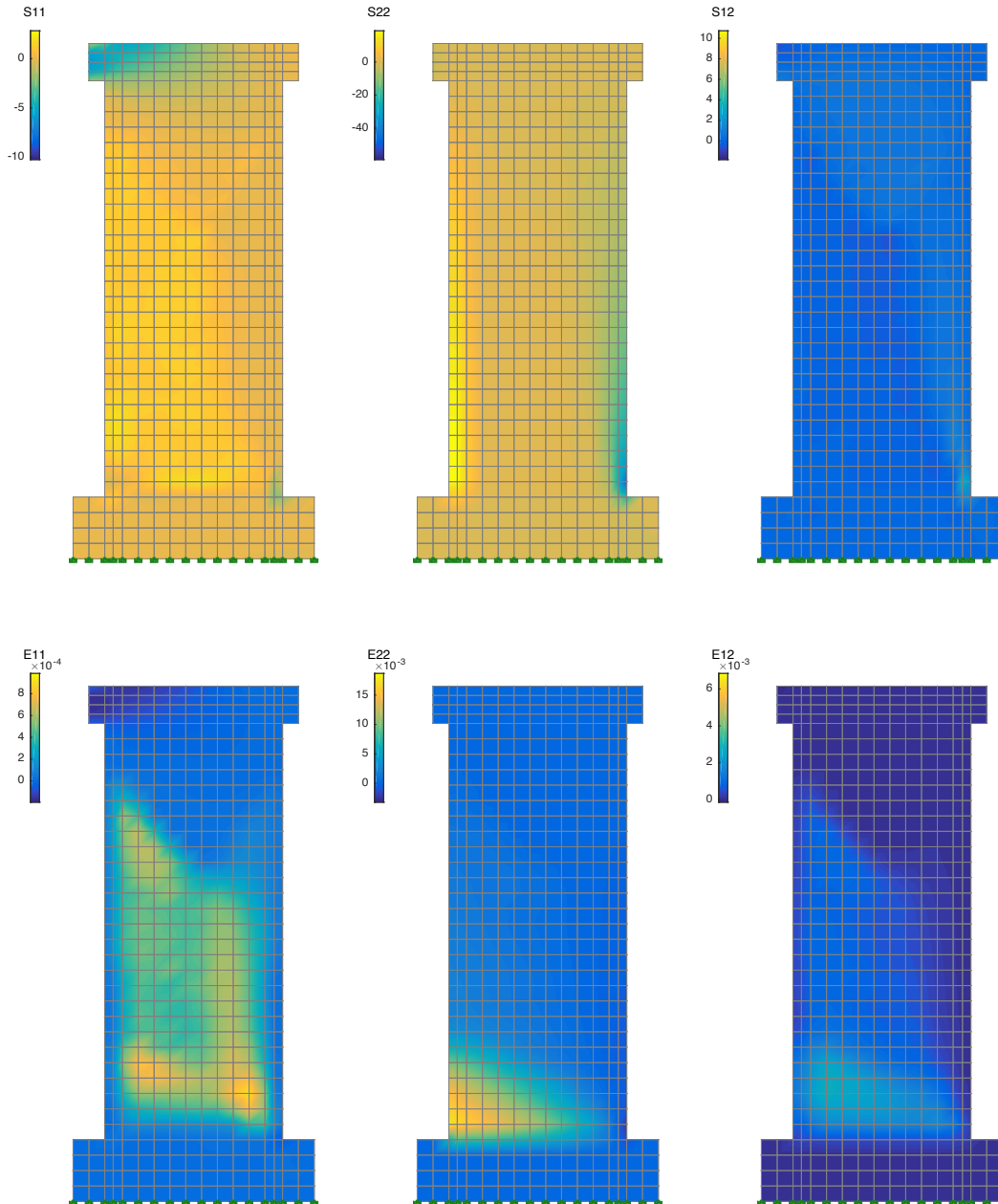


Figura 3.47: Tensión y deformación modelo MR5 al 1% de deriva

Al comparar las distribuciones de esfuerzos se puede ver una similitud en los estados de cargas, debido a que ambos representan desplazamientos positivos, y un notorio aumento de esfuerzos en el caso del desplazamiento máximo.

Para la deformación axial, E11, se aprecia un cambio en la concentración de deformaciones en la zona inferior para el caso de d_{max} . Así mismo, las deformaciones por tracción aumentan considerablemente en la zona inferior izquierda. Un aumento en la escala en los gráficos, sobretodo para E12, muestra un aumento de deformaciones entre el caso de 1% de drift y d_{max} .

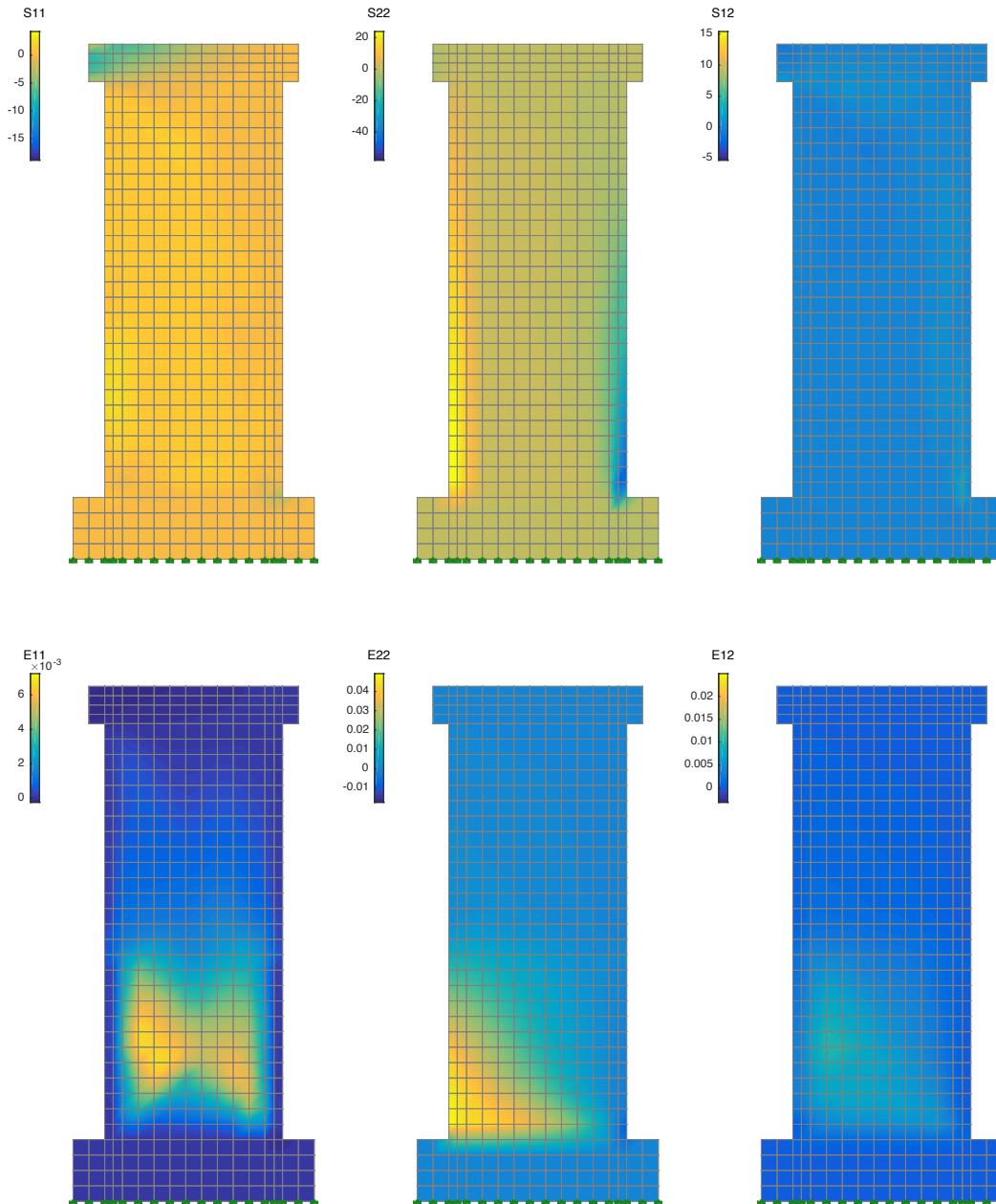


Figura 3.48: Tensión y deformación modelo MR5 al d_{max} - 3.44% de deriva

Capítulo 4

Comparación de Modelos

Los modelos de muros, incluyendo sus resultados y características, nos permiten comparar similitudes y diferencias entre los modelos analizados, las cuales se señalan a continuación.

- El mallado utilizado en todos los modelos fue realizado a partir de una función creada en Matlab, el cual varió para el modelo MR5, en donde la viga de carga debía ser más ancha que la el muro modelado. En todos los modelos, al realizar un mallado con elementos más pequeños se lograba mejorar la predicción del comportamiento obtenido en los resultados. Al añadir elementos de vigas, particionar zonas de mayores tensiones (ej. confinadas) o definir que zona se modelaba como muro y cual como viga de carga permitía variar la dimensión de elementos. Sin embargo, el límite de los elementos estaba cercano a 50 [mm], menos que esto ya no es posible obtener una correcta predicción.
- El uso de los coeficientes de confinamiento formulados por Orakcal et al. [9] dio resultados positivos, obteniendo una correcta predicción de respuesta de los muros experimentales.
- Los coeficientes de daño al recomendados, $\alpha_1 = 0.1 - 0.175$ y $\alpha_2 = 0.5 - 0.6$, fueron reemplazados por coeficientes iguales a $\alpha_1 = 0.1 - 0.15$ y $\alpha_2 = 0.4 - 0.5$. Estos valores fueron determinados al analizar los diferentes modelos y verificar que al rebajar este coeficiente los resultados presentaban menores diferencias con los muros experimentales e incurrían en menos errores de convergencia numérica para desplazamientos elevados.
- Al comparar los diferentes muros, es posible señalar que, en general, aquellos con razones de aspecto mayores a 2, lograban tener un correcto desempeño en todos los ciclos y su modelo no mostraba problemas para converger. Cuando los modelos eran de razón de aspecto más baja a 2, se tenían mayores problemas al converger y reproducir el ciclo de descarga de los diferentes muros experimentales. Estas diferencias en el ciclo de descarga se deben a problemas en la solución numérica, dado que para desplazamientos considerables, sobre fluencia, el ángulo de rotación se acerca a los límites de cuadrante ($\pi/4$, $-\pi/4$, $3\pi/4$ y $-3\pi/4$), provocando errores para definir el comportamiento dominante para esos pasos.

- En la respuesta por flexión y corte, vista en los modelos MR1 y MR2, se obtuvieron resultados con porcentajes de diferencia menores al utilizar la misma metodología que aquella usada experimentalmente. Cabe destacar que para el caso del muro MR1, en donde el muro se separó por pisos a determinada altura, se pierde precisión en la respuesta al comparar resultados al modelar elementos conformados por nodos a cierta altura, se pierde la precisión al buscar comparar con el resultado experimental que puede tener su medidor en un punto que no está ubicado en la misma posición en el modelo. En el caso de MR1 se utilizó un promedio de los valores entre los nodos y en el caso del muro S78, que dividía el muro por tramos, se utilizaron divisiones cercanas a las experimentales.
- La respuesta máxima y la rigidez inicial son los resultados más cercanos a los experimentales, donde en todos los muros se alcanzaron porcentajes de diferencia menores a 5%. El punto de pérdida de capacidad fue correctamente reproducido en todos los muros con excepción del muro MR5, el cual en el último ciclo no logró reproducir la pérdida de capacidad.
- Los gráficos de tensiones y deformaciones para un 1% de drift y para d_{max} , mostraron resultados acordes, presentando un aumento, en ambos casos, para el desplazamiento máximo. Para la deformación E11 se obtuvo, en todos los casos, compresiones en las zonas donde estaba aplicada la carga y en la zona inferior. En E22, una zona a compresión a la derecha inferior y a tracción a la izquierda inferior debido al movimiento del muro, y en E12 una distribución concentrada en la parte inferior. En el caso de las tensiones el modelo mostró similar comportamiento, siendo consecuente con la dirección de la carga.
- En los muros RW2 y WSH6, caso 1 y 3, tenían los valores más altos de s/d , 5.33 y 4.17 respectivamente. Para verificar el pandeo en estos muros, se añadió un modelo de pandeo para las barras de acero embebido. Los resultados no se diferenciaron del resultado con el modelo de acero utilizado en el trabajo, por lo cual no se consideró la respuesta en el presente trabajo. En las investigaciones de estos muros, se señala que existió pandeo solo en los ciclos finales, debido a la pérdida de recubrimiento, luego de reiterados ciclos, siendo consecuente con los resultados obtenidos. Para futuros estudios de muros para estudiar modelos con pandeo, es recomendable estudiar muros que si hayan presentado estos efectos, verificando un elevado s/d , de modo de validar modelos de acero con estos efectos.

Capítulo 5

Recomendaciones de modelo

Luego de analizar los diferentes casos de estudio y realizar diferentes modelos a lo largo de esta investigación para obtener los resultados mostrados en el capítulo 3, es posible señalar recomendaciones para modelar muros de hormigón armado utilizando la plataforma SAFE-TB - "Structural Analysis and Finite Elements ToolBox", desarrollado por Rojas [11]. Estas recomendaciones buscan lograr un acercamiento para representar la estructura en el programa, de modo de obtener resultados representativos al de un muro real y no obtener problemas de convergencia, ni de errores, al modelar o interpretar resultados.

5.1. Mallado

Para lograr un mallado correcto es necesario tener en cuenta el ancho y alto de elementos a lo largo del muro, el pedestal de fundación y la viga de carga.

- **Elementos de Muro:** Se recomiendan anchos y altos de menos de 200-250 [mm] a lo ancho y largo del muro, en especial en las zonas de mayor deformación, como en los casos revisados, la zona inferior. Para la parte confinada también es recomendado utilizar un menor ancho debido a la mayor cuantía de esta zona, dentro de los 100-150 [mm] dependiendo del ancho total confinado del muro. Es recomendado modelar elementos sobre los 50 [mm] para tener una buena representación del hormigón.
- **Pedestal de fundación:** Es recomendado que el ancho de cada elemento sea colineal a los dispuestos en el muro para tener continuidad en los nodos. Para este trabajo se utilizaron 4 elementos verticales del mismo ancho que en el muro.
- **Viga de carga:** Su altura total fue en dependencia de como representar al muro real, aunque cuando se utilizó una altura total, con cuatro elementos, menor a 200 [mm] los resultados no fueron positivos. Es recomendable utilizar un alto por elemento de más de 50-55 [mm].

5.2. Cargas y Nodo de Control

Para disponer las cargas verticales, se recomienda utilizar una distribución de forma equitativa en los nodos superiores, utilizando todos o por zonas. La carga horizontal logra resultados acordes al distribuirla en más de un nodo, de modo de no concretar las fuerzas en un punto y pueden representar mejor el área donde actúan los actuadores hidráulicos.

Recordar que el nodo de control determina el desplazamiento impuesto, por lo tanto, un cambio mostrará distintos comportamientos y fuerzas en los nodos de carga.

5.3. Procedimiento de control de desplazamiento

Para estudiar cargas cíclicas es preferible hacer un análisis preliminar del muro a modelar, el cual consiste en un procedimiento de *push over* hasta el máximo desplazamiento positivo y negativo del protocolo de desplazamiento del análisis cíclico. Para obtener respuestas en menor cantidad de pasos es posible modificar el incremento de desplazamiento Δ_i , obteniendo valores del *slope* ($1/\Delta_i$) de 1/1 a valores de 1/10 o 1/20. Este cambio aumenta el tiempo de solución al disminuir la cantidad de pasos pero perjudica la precisión del resultado.

5.4. Material

5.4.1. Hormigón

Utilizar el f_c indicado por los ensayos. En caso contrario utilizar el mismo que al diseñar, aunque este normalmente es menor al real obtenido por los ensayos. Es recomendable aumentar este valor en un 10% para representar el valor esperado del hormigón. El resto de los coeficientes utilizados para modelar el comportamiento uniaxial del hormigón aparece en detalle en el capítulo 2.

5.4.2. Acero

Colocar las propiedades del material según se indican en los ensayos de laboratorio. De no poseer, $E = 200,000[MPa]$ es un valor representativo para el esfuerzo de fluencia según sea el utilizado al diseñar.

5.5. Coeficiente por confinamiento

En un primer acercamiento, se recomienda iniciar el análisis con un valor de $\beta_{conf} = 1.1$. Posteriormente, se utiliza el resultado según la ecuación formulada por Orakcal [9] en 2006, y comparar su efectividad y cercanía de resultado, lo cual en este trabajo presento una mejora en la predicción de resultados.

5.6. Coeficientes de daño

En un primer análisis no es recomendado utilizar daño en el concreto, ni en el acero. Es mejor ver como se comporta el modelo de muro sin daños al hormigón y acero, corregir errores en el mallado del elemento o de las fuerzas impuestas, para luego comenzar con los coeficiente a_1 y a_2 para daños en el acero, modificando su valor de 0 a -0.002 , y posteriormente a realizar un análisis con esta modificación, realizar el cambio en los coeficientes de daño del concreto.

El daño en el concreto debe iniciar en lo recomendado según el capítulo 2. En este trabajo se han visto mejores resultados al utilizar valores de $\alpha_1 = 0.1 - 0.15$ y $\alpha_2 = 0.4 - 0.5$, los cuales son menores a los recomendados. En caso de presentarse problemas de convergencia al imponer daño, realizar cambios en los coeficientes de daño.

5.7. Definición de elementos

Para definir el elemento al modelo se debe procurar generar correctamente aquellos que serán estudiados y aquellos que solo sean de traspaso de carga o de condiciones de borde. Tener especial cuidado al agregar elementos adicionales al modelo, como por ejemplo vigas, debido a que estas probablemente aumenten la resistencia del muro pero pueden existir mayores errores de convergencia en las uniones de estas al muro. Siempre intentar recrear de manera cercana y fiel las propiedades y dimensiones de los muros que se desean modelar.

Capítulo 6

Conclusión

El presente trabajo tuvo como objetivo comparar las respuestas analíticas de modelos realizados en la plataforma programada en Matlab, SAFE-TB - "Structural Analysis and Finite Elements ToolBox", desarrollada por Rojas [11], para muros de hormigón armado ante cargas cíclicas y estáticas, con respuestas experimentales. Para este fin, se modelaron 5 muros de hormigón armado y se sometieron a los mismos protocolos de carga que los muros experimentales, con el fin de comparar su comportamiento y lograr validar el uso de la plataforma para modelar M.H.A. Además, analizando la modelación realizada, se puede estudiar la distribución de esfuerzos en los diferentes casos, las rigideces iniciales y de fluencia, y comportamientos globales del muro.

En cada muro modelado se estudio la respuesta ante desplazamientos impuestos, obteniendo como resultado la fuerza necesaria para producir el desplazamiento requerido y comparar estas con su muro experimental correspondiente. Además, se estudiaron los desplazamientos por corte y flexión en aquellos casos experimentales con estos desplazamientos estudiados, replicando el método de cálculo empleado con las investigaciones correspondientes, y por medio de la deformación angular de los elementos. En ambos casos revisados, MR1 y MR2, el resultado por medio del método empleado en el ensayo experimental mostró resultados más representativos a los resultados experimentales que los otros utilizados.

Se realizo una envolvente en todos los resultados de los modelos, las cuales fueron comparadas con la envolvente de los muros experimentales, estudiando tanto la ubicación de los puntos (desplazamiento y fuerza) como la pendiente de la recta entre estos puntos, promediando entre el resultado para desplazamientos positivos y negativos, y comparando esta rigidez por tramos.

La facilidad de representar los diferentes casos modelados por medio de elementos cuadriláteros tipo membrana permitieron una correcta representación de cada muro experimental, logrando replicar las dimensiones y comportamientos en las diferentes zonas del mallado, sobretodo en las zonas confinadas.

El efecto de pandeo en las barras longitudinales fue incorporado en el modelo MR1 y MR3, sin obtener diferencias en los resultados al compararlo con el modelo de acero utilizado sin

pandeo, por lo cual no se utilizaron estos resultados para comparar la respuesta de estos dos modelos. Estos dos casos fueron los seleccionados para estudiar el pandeo debido a que eran los muros de razón s/d más alto (5.33 y 4.17 respectivamente). La posible razón de no presentar diferencias se puede deber a que ninguno de estos muros, en sus reportes experimentales, se señala que hayan presentado casos de pandeos a lo largo del ensayo, si no cuando el muro ya había fallado por los reiterados ciclos y aplastamiento de sus zonas confinadas, sumado a la pérdida de recubrimiento, y no por un alto espaciamiento o barras en el confinamiento de bajo diámetro.

Todos los valores de parámetros y coeficientes utilizados en la modelación se basarán en el trabajo de F. Rojas [12], los cuales lograron predecir el comportamiento de los muros experimentales, permitiendo una respuesta correcta del comportamiento en los ciclos de carga y descarga, como en su resistencia máxima. El mallado realizado es importante al usar la plataforma para modelar los muros, tanto las dimensiones de los elementos para cada zona como las dimensiones de las distintas zonas de cada muro. Delimitar el pedestal de fundación, la viga de carga, y sobretodo, el muro en estudio, de forma que represente al caso estudiado permite predecir su comportamiento. Además, el uso de elementos de menor dimensiones para zonas con altos esfuerzos, como el confinamiento, influyen directamente en la correcta modelación del muro.

Para cada muro experimental estudiado se realizó un análisis de resultados, mostrado en la sección *Análisis y Resultado* de cada caso, en el capítulo 3. Al revisar los diferentes resultados se puede concluir que los modelos realizados con elementos membranas, todos en dos dimensiones, logran predecir el comportamiento de los muros experimentales, representando su comportamiento global, desplazamiento y fuerzas, rigidez inicial y de fluencia, además de los desplazamientos por flexión y por corte, al igual que los ciclos de carga. Diferencias en los ciclos de descarga se presentan solo fuerzas cercanas a 0 y con desplazamientos elevados, las cuales se deben a un problema de solución numérica con el ángulo de rotación del modelo de hormigón, en aquellos muros con comportamiento de muro de corte, caso MR2 y MR4.

En los modelos MR4 y MR5 se muestran las mayores diferencias, las cuales son atribuidas por la dificultad de representar estos dos muros. En el caso del muro MR4, al agregar las losas elásticas y añadir los coeficientes de daño del hormigón, se presentaron los errores de convergencia del modelo, lo cual se solucionó por medio de rebajar los coeficientes de daño, mostrando un correcto comportamiento global. En el modelo MR1, las complicaciones experimentales por los efectos de pandeo global desde los ciclos iniciales, además de la adición de soporte transversal, no permiten replicar el punto de fluencia, ni el degradamiento en su último ciclo, para un drift de 3.44 %. Sin embargo, tanto la rigidez inicial como la capacidad máxima si representan al muro experimental.

La comparación entre modelos nos permite ver una relación entre la razón de aspecto con el comportamiento de los muros, debido a que en aquellos con razón menor, presentan leves errores numéricos para valores de la fuerza cercanos a 0 durante los ciclos de descarga, solo presentes para los ciclos con desplazamientos elevados, sobre un 1 % de drift. En los diferentes modelos se definieron coeficientes de daño dentro del rango $\alpha_1 = 0.1 - 0.15$ y $\alpha_2 = 0.4 - 0.5$, concluyendo en utilizar valores menores a los recomendados. Además, en todos los modelos obtenemos igualdad de comportamientos para las distribuciones de esfuerzos y deformaciones,

siendo acordes al desplazamiento positivo impuesto. Finalmente, de esta comparación se determina que los resultados de rigidez inicial y de capacidad máxima fueron las predicciones de los diferentes modelos más cercanas, con un porcentaje de diferencia menor al 5%.

Las recomendaciones de modelo, utilizando la plataforma, entregan información y herramientas para realizar un modelo correcto, con capacidad de predecir el comportamiento de muros por medio de elementos membrana, reduciendo los errores al modelar.

Glosario

A_g	: Área bruta de acero
α_i	: Coeficiente de Daño
@	: Intervalo de disposición
β_{conf}	: Coeficiente de confinamiento
b	: Valor propuesto experimental
B	: Factor de cambio de f_y
DM	: Doble Malla de refuerzo
E_s	: Modulo de Young del acero
E_{c0}	: Rigidez inicial del concreto
ε_c	: Deformación a compresión
ε_{c0}	: Deformación a máximo esfuerzo en compresión
ε_{cr}	: Deformación a máximo esfuerzo en tracción
ε_m^c	: Máxima deformación anterior registrada a compresión
ε_m^t	: Máxima deformación anterior registrada a tracción
ε_p^c	: Deformación plástica a compresión
ε_p^t	: Deformación plástica a tracción
f_c'	: Resistencia mínima a compresión del hormigón
f_{cr}	: Resistencia máxima a compresión del hormigón
f_y	: Esfuerzo de fluencia
f_{yv}	: Esfuerzo de fluencia de barras verticales
f_{yh}	: Esfuerzo de fluencia de barras horizontales
f_{yconf}	: Esfuerzo de fluencia de barras verticales en zona confinada
f_{ys}	: Esfuerzo de fluencia de barras del estribos
ϕ	: Diámetro nominal
$\phi_{estribos}$: Diámetro nominal de estribos
γ_h	: Peso específico del hormigón
H_w	: Altura de muro
H_w/L_w	: Razón de aspecto de muro
k	: Valor calibrado experimental
L_w	: Largo de muro
L_w/T_w	: Razón de esbeltez de muro
L_{conf}	: Largo de confinamiento
N^o	: Número de barras
ν	: Coeficiente de Poisson
n	: Valor calibrado experimental

$\frac{P}{A_g \cdot f'_c}$: Porcentaje de carga por área
r_c	: Recubrimiento de hormigón
ρ_{conf}	: Cuantía vertical en zona de confinamiento
$\rho_{estribos}$: Cuantía horizontal de los estribos
ρ_h	: Cuantía horizontal
ρ_v	: Cuantía vertical
ρ_s	: Cuantía en por estribos
s	: Espaciamiento entre barras de estribos
s/d	: Razón espaciado con diámetro de barra vertical en confinamiento
σ	: Esfuerzo axial
σ_{c_j}	: Esfuerzo en la dirección normal i
T_w	: Espesor de muro
τ	: Esfuerzo de corte

Bibliografía

- [1] BELARBI, A., AND HSU, T. Constitutive laws of concrete in tension and reinforcing bars stiffened by concrete. *ACI Structural Journal*, 91(4) (1994), 465–474.
- [2] CARREIRA, D., AND KUANG-HAN, C. Stress-strain relationship for plain concrete in compression. *ACI Structural Journal*, 82(6) (1985), 797–804.
- [3] DAZIO, A., BEYER, K., AND BACHMANN, H. *Engineering Structures.*, vol. 31. ELSEVIER, Ámsterdam, Países Bajos., 2009.
- [4] FILIPPOU, F., POPOV, E., AND BERTERO, V. Effect of bond deterioration on hysteretic behavior of reinforced concrete joints. ucbl/eerc 83/19; earthquake engineering research center, university of california, berkeley, 1983.
- [5] MASSONE, L. *RC Wall Shear - Flexure Interaction: Analytical and Experimental Responses.* PhD thesis, University of California, Los Angeles, Los Angeles, CA, USA., 2006. Department of Civil and Environmental Engineering.
- [6] MASSONE, L., AND MORODER, D. Buckling modeling of reinforcing bars with imperfections. *Engineering Structures*, 31 (3) (2009), 758–767.
- [7] MENEGOTTO, M., AND PINTO, P. Method of analysis of cyclically loaded reinforced concrete plane frames including changes in geometry and non-elastic behavior of elements under combined normal force and bending. in: IABSE symposium on the resistance and ultimate deformability of structures acted on by well-defined repeated loads, lisbon, 1973.
- [8] OESTERLE, R., FIORATO, A., ARISTIZABAL-OCHOA, J., AND CORLEY, W. Hysteretic response of rc structural walls. Special Publication (63): 243-274., 1980.
- [9] ORAKCAL, K., MASSONE, L., AND WALLACE, L. Analytical modeling of reinforced concrete walls for predicting flexural and coupled-shear-flexural responses. peer report. Tech. rep., Pacific Earthquake Engineering Research Center. University of California., Los Angeles, CA, USA, 2006.
- [10] PALERMO, D., AND VECCHIO, F. Compression field modeling of reinforced concrete subjected to reversed loading: Formulation. *ACI Structural Journal*, 100(5) (2003), 616–625.

- [11] ROJAS, F. *Development of a nonlinear quadrilateral layered membrane element with drilling degrees of freedom and a nonlinear quadrilateral thin flat layered shell element for the modeling of reinforced concrete walls*. PhD thesis, University of Southern California, Los Angeles, CA, USA., 2012. Department of Civil and Environmental Engineering.
- [12] ROJAS, F., ANDERSON, J., AND MASSONE, L. A nonlinear quadrilateral layered membrane element with drilling degrees of freedom for the modeling of reinforced concrete walls. Santiago, Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas., 2016.
- [13] THORENFELDT, E., TOMASZEWICZ, A., AND JENSEN, J. Mechanical properties of high-strength concrete and application in desing. In: Symposium Utilization of High-Strength Concrete. Stavanger, Norway, 1987.
- [14] TRAN, T. *Experimental and Analytical Studies of Moderate Aspect Ratio Reinforced Concrete Structural Walls*. PhD thesis, University of California, Los Angeles, Los Angeles, CA, USA., 2012. Department of Civil and Environmental Engineering.
- [15] VALLENAS, J., BERTERO, V., AND POPOV, E. Hysteretic behavior of reinforced concrete structural walls. Tech. rep., University of California Berkeley, Earthquake Engineering Research Center., Auckland, CA, USA., 1979.
- [16] WALLACE, J., AND THOMSEN, J. Displacement-based design of reinforced concrete structural walls: An experimental investigation of walls with rectangular and t-shaped cross-sections. Tech. rep., Clarkson University, Department of Civil and Environmental Engineering., Richmond, CA, USA, 1995.
- [17] YAÑEZ, F. *Seismic behaviour of r.c. walls with irregular openings*. PhD thesis, University of Canterbury, Canterbury, New Zealand, 1993.

Apéndice A

Figuras

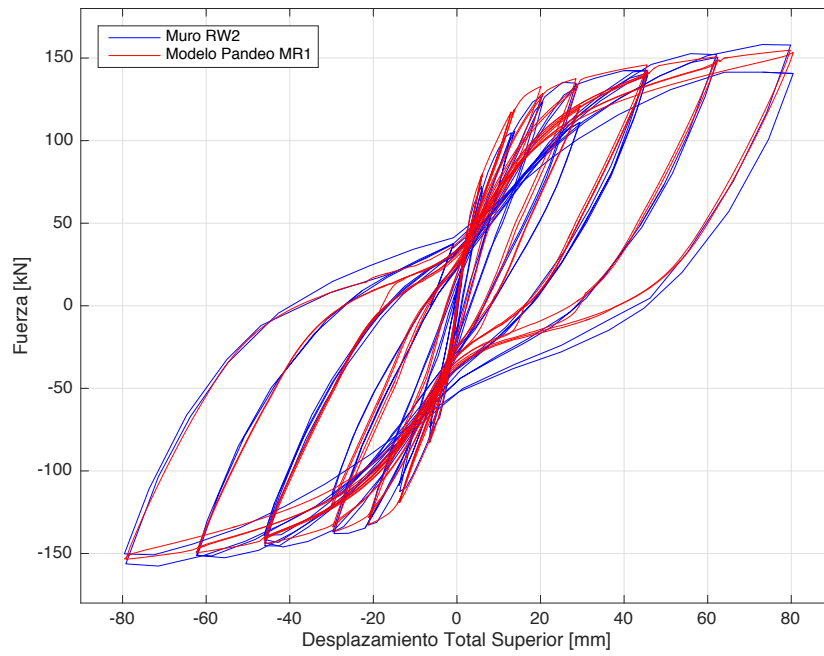


Figura A.1: Ciclo Carga-Descarga para muro RW2 y MR2 con modelo de pandeo

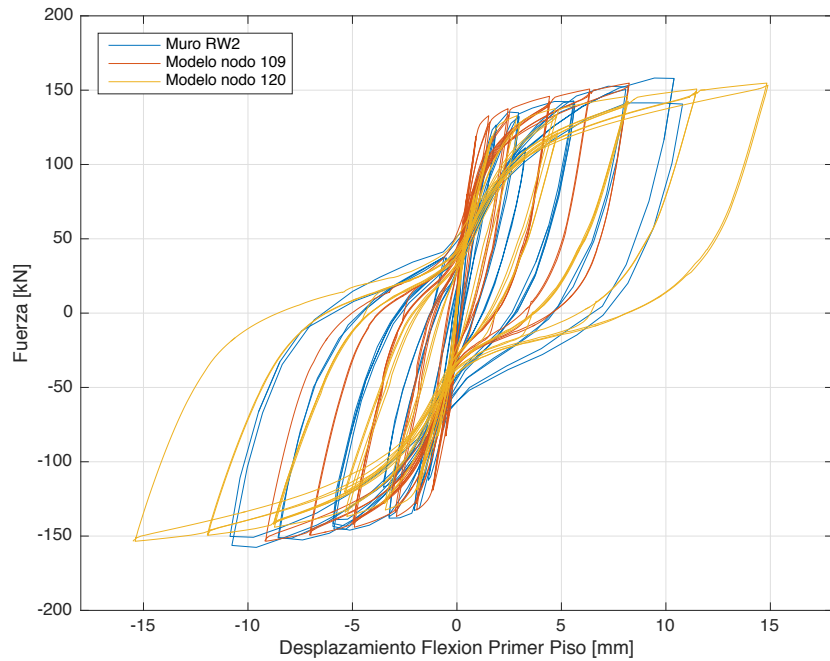


Figura A.2: Desplazamiento por flexión para modelo MR1, Resultado con nodo 109 y 120 como altura de primer piso.

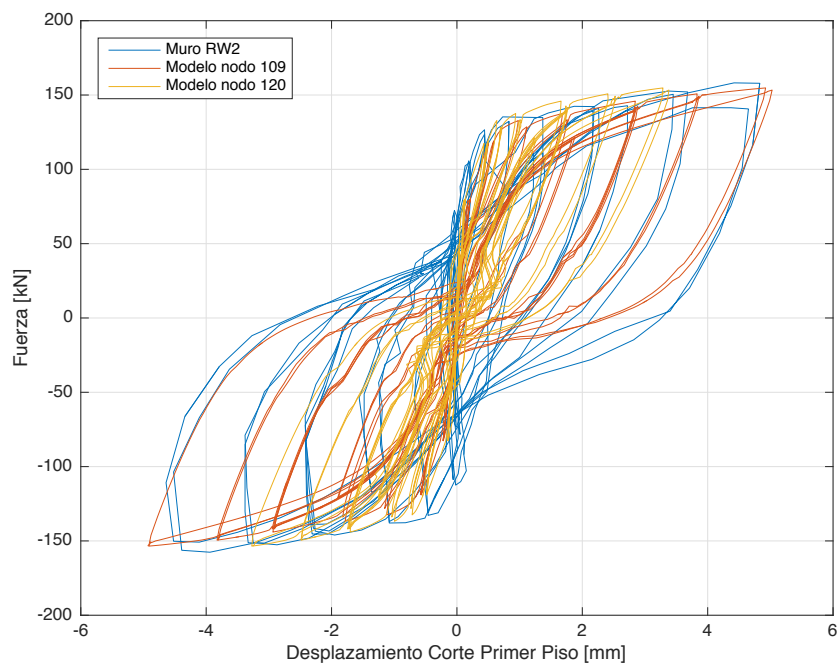


Figura A.3: Desplazamiento por corte para modelo MR1, Resultado con nodo 109 y 120 como altura de primer piso.

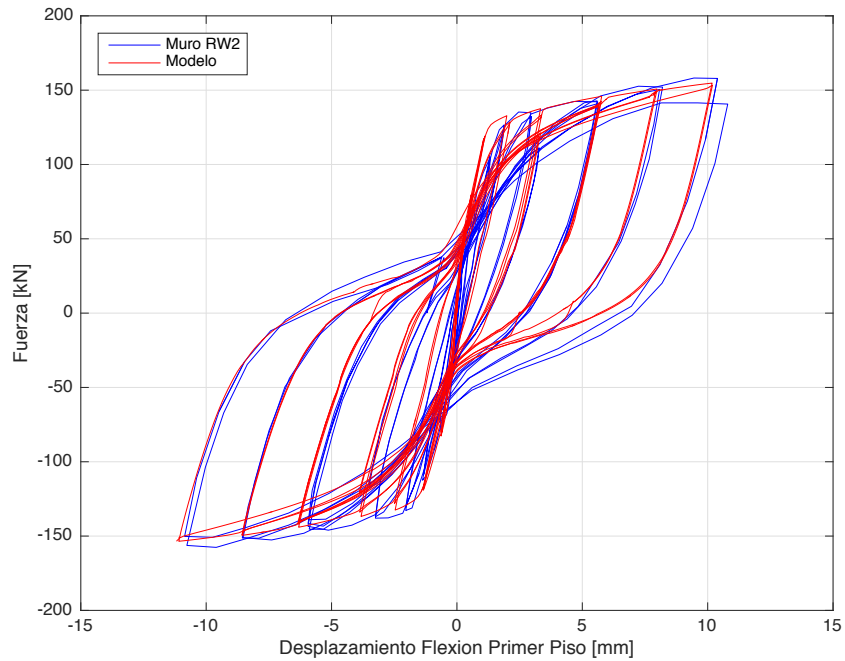


Figura A.4: Desplazamiento por flexión para modelo MR1, utilizando método por deformación angular del elemento.

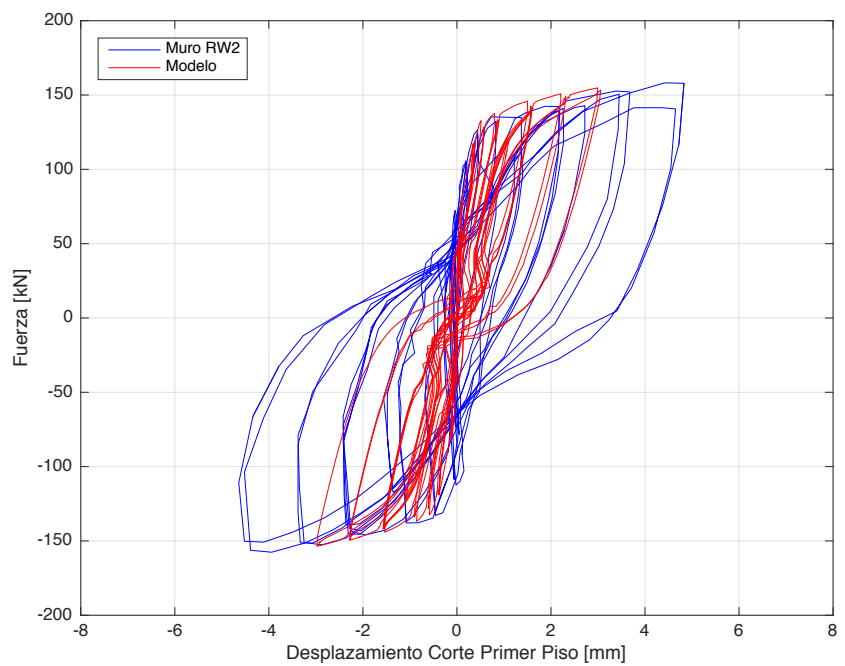


Figura A.5: Desplazamiento por corte para modelo MR1, utilizando método por deformación angular del elemento.

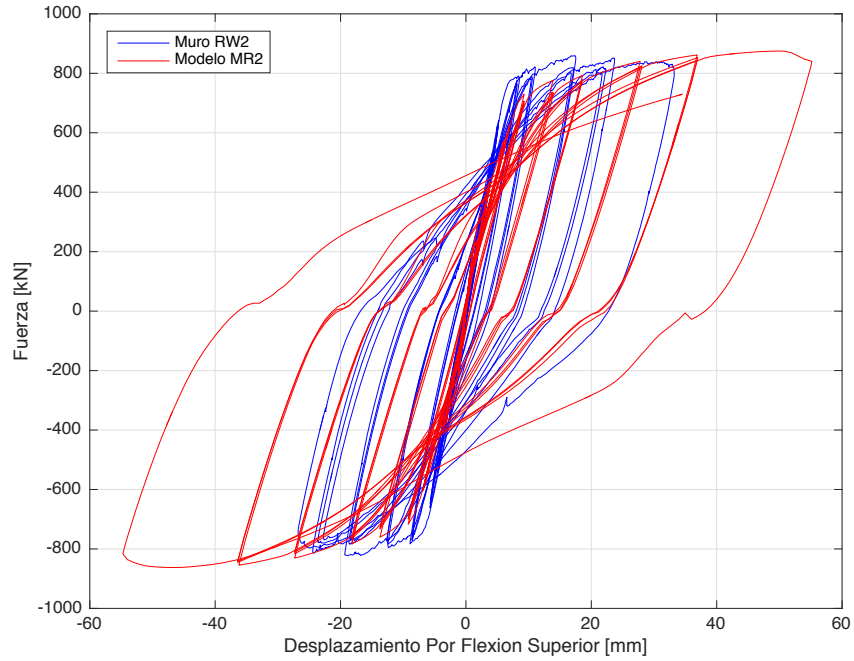


Figura A.6: Desplazamiento por flexión para modelo MR2, utilizando método por deformación angular del elemento.

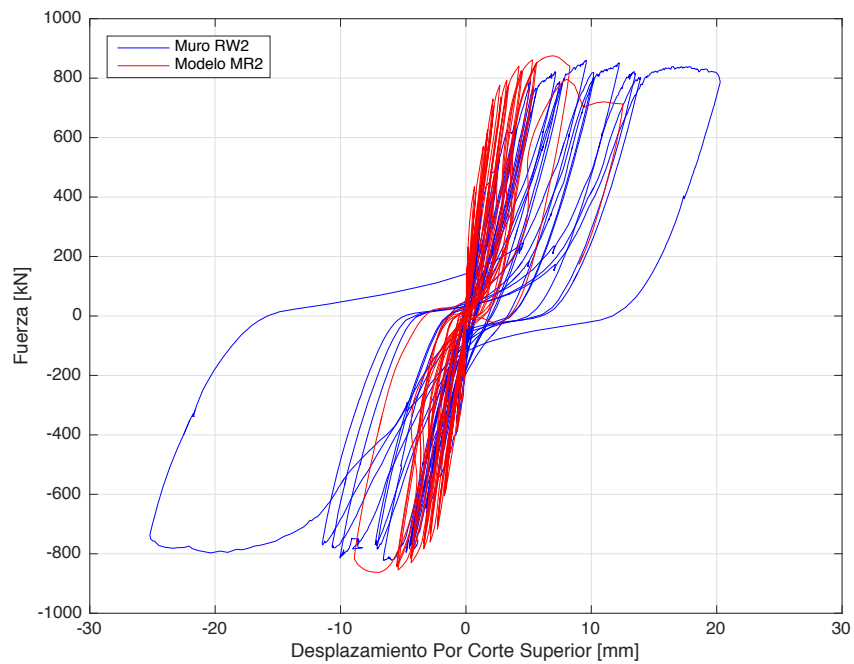


Figura A.7: Desplazamiento por corte para modelo MR2, utilizando método por deformación angular del elemento.

Apéndice B

Códigos de Programación

Código MR1

Modelo

```
tic
%% Create the Model
modelObj=Model();

nodes2D = load('Nodes_RW2_RW.txt');
numberNodes = length(nodes2D);
elements2D = load('Elements_RW2_RW.txt');
numberElements = size(elements2D,1);
nodesSPC = 1 : 15;
horizontalNodeLoads = [ 263 274 285 ];
verticalNodeLoads = 297 : 305;
nodeTagToMonitorInTheDisplacementControl = 252;
nodes_shearDisp = [63 73 120 130];
foundationElements = 1 : 56;
superiorElements= 227 : 266;
boundaryElements = [57 : 58 65 : 66 67 : 68
                    75 : 76 77 : 78 85 : 86 87 :
                    88 95 : 96 97 : 98 105 : 106
                    107 : 108 115 : 116 117 : 118 125
                    : 126 127 : 128 135 : 136 137 : 138
                    145 : 146 147 : 148 155 : 156 157 :
                    158 165 : 166 167 : 168 175 : 176
                    177 : 178 185 : 186 187 : 188 195
                    : 196 197 : 198 205 : 206 207 : 208
                    215 : 216 217 : 218 225 : 226];
centerElements = [59 : 64 69 : 74
                  79 : 84 89 : 94
                  99 : 104 109 :
                  114 119 : 124
                  129 : 134 139 : 144
                  149 : 154 159 : 164
                  169 : 174 179 : 184
                  189 : 194 199 :
                  204 209 : 214
                  219 : 224];
```



```

%% Create the Nodes
if optionTypeElement == 1
    numDOF = 2;
else
    numDOF = 3;
end % if

nodesObjects = cell(numberNodes,1);
for i = 1:numberNodes
    nodesObjects {i} = Node(i,numDOF,nodes2D(i,2:3));
    % Add the Nodes that define the element
    modelObj.addNode(nodesObjects {i});
end % for i

%% Material Properties
magnitudGSelfWeightLoad = 9800; % [mm/s^2]
rhoConcreteMaterial = 24.5*(10^-6)/magnitudGSelfWeightLoad; % [N/mm^3] Concrete
nuConcrete = 0.2;

% Steel
%Be Steel - "Confined"
Es_BeVertical = 200000; % [MPa]
Fy_BeVertical = 434; % [MPa]
d_BarBeVertical = 9.525; % [mm]

%Vertical Steel
Es_Vertical = 200000; % [MPa]
Fy_Vertical = 448; % [MPa]

%Horizontal Steel
Es_Horizontal = 200000; % [MPa]
Fy_Horizontal = 448; % [MPa]

%Hoops Steel
Es_Hoops = 200000; % [MPa]
Fy_Hoops = 434; % [MPa]

% We add a parameter to the Name of the File to save
nameFileMat = [ nameFileMat '_WR' num2str(optionSpecimenWall) ];

%% Geometry of Wall
% Dimension Rectangular Wall

height_Wall = 3660; % [mm]
thickness_Wall = 102; % [mm]
coverConcrete_Wall = 9; % [mm]
lengthBoundary_Wall = 191; % [mm]
lengthCenter_Wall = 838; % [mm]

```

```

% We Set the Total Horizontal Force use to Analysis with Displacement
% Control
totalHorizontalForceLoad = 1000; % [N]

% We define the Total Vertical Load
totalVerticalForceLoad = -378099; % [N]

%% %% DEFINIDO POR ENSAYOS
% We define the Lateral Drift Procedure
lateralDrift = [0.0    -0.2 2.3 -2.6  2.5   -2.4  5.9   -6.5  6.1   -6.5  13.0  -13.7
               13.8  -13.8  20.1  -21.2  20.6  -21.1  28.5  -29.4  28.8  -29.7  45.5  -46.0
               45.7  -46.1  29.5  -29.8  29.3  -30.0  45.6  -46.1  45.3  -46.0  62.1  -62.1
               62.5  -62.3  79.8  -79.2  80.5  -79.6]; % ciclo
%% %% DEFINIDO POR ENSAYOS
%% %%

numberLateralDrift = length(lateralDrift);
time = zeros(numberLateralDrift,1);
numberStepsPerCicle = zeros(numberLateralDrift,1);
totalTime = 0;
slope = 1/1;
for i=1:numberLateralDrift
    lateralDrift(i) = sign(lateralDrift(i))*ceil(abs(lateralDrift(i))*10)/10;
end % for i
for i=1:numberLateralDrift-1
    totalTime = totalTime + ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
    time(i+1) = totalTime;
    numberStepsPerCicle(i+1) = ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
end % for i
fprintf('Number of Steps to Perform is: %s \n',num2str(totalTime));

factorReductionFc = 1.1;

%% %% Confinamiento

% Concrete
Fc_Pedestal = -45; % [MPa]
Ec_Pedestal = 4700*sqrt(-Fc_Pedestal); % [MPa]

Fc = -43*factorReductionFc; % [MPa] 43
strainAtFc_Center = -0.003; % [mm/mm]
strainAtFc_Boundary = -0.0055; % [mm/mm]
Fcr = 0.31*sqrt(-Fc); % [MPa]
strainAtFcr = 0.00008; % [mm/mm]

%% %% Cuantias de Armaduras

```

```

rho_Hoops_Area = 0.0069;
rho_BeVerticalSteel_Area = 0.0293;
rho_HorizontalSteel_Area = 0.0033;
rho_VerticalSteel_Area = 0.0030;

%% Steel Constant

ratioRhoFyoverFc = abs(rho_Hoops_Area*Fy_Hoops/Fc);
Kconf_Opt1 = 1 + ratioRhoFyoverFc; % Orakcal et al. 2006
Kconf_Opt2 = 1 + 0.92*ratioRhoFyoverFc - 0.76*ratioRhoFyoverFc^2; % Vecchio 1992 (Kupfer et
al 1969)
Kconf_Opt3 = -1.254 + 2.254*sqrt(1+7.94*(ratioRhoFyoverFc)) - 2*(ratioRhoFyoverFc); % Mander
et al (1988)
Kconf = Kconf_Opt1; % Use

%% Create the 2D Material
% Create the Smearred Steel 2D Material
% Steel 2D Material Class Parameters:

if optionSteel == 1
    nameFileMat = [ nameFileMat '_SES' ];
    bsh = 0.01;
    Ac = 1.9;
    Rc = 10;
    OrientationEmbeddedSteel = 0;

    smearedESteel_Area_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerROT2DMaterial01('SmearedESteel_Area',[Es_Horizontal,Es_
    Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[2*rho_HorizontalSteel_Area,2*rho_VerticalSteel_Ar
    ea],OrientationEmbeddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Center,0);
    smearedESteel_AreaBe_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerROT2DMaterial01('SmearedESteel_AreaBe',[Es_Horizontal,Es
    _BeVertical],[(Fy_Horizontal*rho_HorizontalSteel_Area +
    Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Fy_BeVertical],[Fcr,F
    cr],[2*(rho_HorizontalSteel_Area+rho_Hoops_Area),2*rho_BeVerticalSteel_Area],OrientationEmb
    eddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Boundary,0);

elseif optionSteel == 2
    nameFileMat = [ nameFileMat '_SS' ];
    OrientationEmbeddedSteel = 0;
    R0 = 18;
    CR1 = 0.9;
    CR2 = 0.15;
    a1 = -0.002; %To model Damage in the steel
    a2 = 1;
    a3 = -0.002; %To model Damage in the steel
    a4 = 1;

```

```

    smearedESteel_Area_2DMatObj =
    SmearedEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearedESteel_Area',[Es_Horizontal,Es_
    Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_
    Area,rho_VerticalSteel_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,str
    ainAtFc_Center,0);
    smearedESteel_AreaBe_2DMatObj =
    SmearedEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearedESteel_AreaBe',[Es_Horizontal,Es
    _BeVertical],[(Fy_Horizontal*rho_HorizontalSteel_Area +
    Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Fy_BeVertical],[Fcr,F
    cr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_Area+rho_Hoops_Area,rho_BeVerticalSte
    l_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,strainAtFc_Boundary,0);

```

```

elseif optionSteel == 3

```

```

    nameFileMat = [ nameFileMat '_SSP' ];
    OrientationEmbeddedSteel = 0;
    Bsh=0.01;
    R0 = 18;
    CR1 = 0.9;
    CR2 = 0.15;
    a1 = -0.002; %To model Damage in the steel
    a2 = 1;
    a3 = -0.002; %To model Damage in the steel
    a4 = 1;

```

```

    M=38;
    Ecase = 1/300;
    Ehtens = 0.04;
    Eutens = 0.11;
    Sigutens = 630;
    Eohtens = 7000;
    Ef =0.55;
    Ff =0;
    InitialStress = 0;

```

```

steel1UniaxialLayer_Area=SteelUniaxialMaterial('SteelUniaxial_Area',Es_Vertical,Fy_Vertical,Bsh,R0
,CR1,CR2,a1,a2,a3,a4);

```

```

steel2UniaxialLayer_Area=SteelUniaxialMaterial('SteelMMUniaxial_Area',Es_Horizontal,Fy_Horizo
ntal,Bsh,R0,CR1,CR2,a1,a2,a3,a4);

```

```

steel1UniaxialLayer_AreaBe=SteelUniaxialMaterial('SteelUniaxial_AreaBe',Es_BeVertical,(Fy_Horiz
ontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Bsh,R0,CR1,CR2,a1,a
2,a3,a4);

```

```
steel2UniaxialLayer_AreaBe=SteelMMUniaxialMaterial('SteelMMUniaxial_AreaBe',M,d_BarBeVertical,76,Ecase,440,Es_BeVertical,Ehtens,Eutens,Sigutens,Eohtens,Ef,Ff,InitialStress);
```

```
    smearedESteel_Area_2DMatObj =
SteelDoubleLayer2DMaterial('SmearedSteel_Area',rho_HorizontalSteel_Area,steel1UniaxialLayer_Area,rho_VerticalSteel_Area,steel2UniaxialLayer_Area,OrientationEmbeddedSteel);
    smearedESteel_AreaBe_2DMatObj =
SteelDoubleLayer2DMaterial('SmearedSteel_AreaBe',rho_HorizontalSteel_Area+rho_Hoops_Area,steel1UniaxialLayer_AreaBe,rho_BeVerticalSteel_Area,steel2UniaxialLayer_AreaBe,OrientationEmbeddedSteel);
```

```
end
```

```
%% Create the Concrete Uniaxial Material
```

```
if optionDamage == 0
    damageConstantUnconf_1 = 0;
    damageConstantUnconf_2 = 0;
    damageConstantConf_1 = 0;
    damageConstantConf_2 = 0;
elseif optionDamage == 1
    damageConstantUnconf_1 = 0.1;
    damageConstantUnconf_2 = 0.5;
    damageConstantConf_1 = 0.1;
    damageConstantConf_2 = 0.5;
end % if optionConcrete
```

```
%% Create the Orthotropic Rotating Angle 2D Material
```

```
% Tension Stiffening Concrete 2DMaterial
nameFileMat = [ nameFileMat '_TS' ];
K = 1;
AlphaC = 0.32;
AlphaT = 0.08;
```

```
% Unconfined
```

```
R = 3.1;
B = 0.6;
orthotropicConcrete_Area_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_Area',Fc,strainAtFc_Center,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstantUnconf_1,damageConstantUnconf_2,rhoConcreteMaterial);
```

```
% Confined
```

```
R = 2.1;
B = 0.4;
orthotropicConcrete_AreaBe_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_AreaCo'
```

```
,Kconf*Fc,strainAtFc_Boundary,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstantConf_1,damageConstantConf_2,rhoConcreteMaterial);
```

```
if optionDamage == 0
elseif optionDamage == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat 'wUD_1_5' ];
end % if optionConcrete
```

```
if optionSelfWeight == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_SW' ];
end % if
```

```
%%% Profile Properties
% We create the profile for the Reinforced Concrete Wall
coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_Area_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_Area',coordinatesConcreteLayers,c
ordinatesReinforcedSteelLayers,thickness_Wall);
```

```
coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_AreaConf_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_AreaBe',coordinatesConcreteLayer
s,coordinatesReinforcedSteelLayers,thickness_Wall);
```

```
%Create the Reinforced Concrete Layered Membrane Section
concrete2DMaterialObjectAtEachLayer = { orthotropicConcrete_Area_RA2DMatObj };
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_Area_2DMatObj };
rcWall_Area_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_Area',rcWall_Area_ProfileObj,concr
ete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);
```

```
concrete2DMaterialObjectAtEachLayer = { orthotropicConcrete_AreaBe_RA2DMatObj };
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_AreaBe_2DMatObj };
rcWall_AreaConf_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_AreaConf',rcWall_AreaConf_Profile
Obj,concrete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);
```

```
%Create the Elastic Concrete Membrane Pedestal Section
```

```
% Create the 2D Material
elasticConcrete2DMat =
ElasticIsotropicPlaneStress2DMaterial('ElasticConcrete2DMat',Ec_Pedestal,nuConcrete,rhoConcret
eMaterial);
thickness_Pedestal = thickness_Wall*5; % [mm]
```

```

rcWall_Pedestal_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_Pedestal);
rcWall_Pedestal_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_Pedestal_ProfileObj,elasticConcrete2DMat);

%Create the Elastic Concrete Membrane Superior Section
%Create the 2D Material

thickness_SupSection = thickness_Wall; % [mm]
rcWall_SupSection_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_SupSection);
rcWall_SupSection_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_SupSection_ProfileObj,elasticConcrete2DMat);

%% Create the Element
quad4NObjects = cell(numberElements,1);
if optionTypeElement == 2
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_Rot' ];
    % Elements with Drilling DOF

    % We add the Foundation Elements
    numberElements_Area = length(foundationElements);
    for i = 1:numberElements_Area
        node1 = elements2D(foundationElements(i),2);
        node2 = elements2D(foundationElements(i),3);
        node3 = elements2D(foundationElements(i),4);
        node4 = elements2D(foundationElements(i),5);
        quad4NObjects {foundationElements(i)} = Quad4NR02(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj,optionTypeElement);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {foundationElements(i)});
    end % for i

    % We add the Superior Elements
    numberElements_Area = length(superiorElements);
    for i = 1:numberElements_Area
        node1 = elements2D(superiorElements(i),2);
        node2 = elements2D(superiorElements(i),3);
        node3 = elements2D(superiorElements(i),4);
        node4 = elements2D(superiorElements(i),5);
        quad4NObjects {superiorElements(i)} = Quad4NR02(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj,optionTypeElement);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {superiorElements(i)});
    end % for i

```

```

% We add the Boundary Elements
numberElements_Area = length(boundaryElements);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements(i),2);
    node2 = elements2D(boundaryElements(i),3);
    node3 = elements2D(boundaryElements(i),4);
    node4 = elements2D(boundaryElements(i),5);
    quad4NObjects {boundaryElements(i)} = Quad4NR02(['E'
num2str(boundaryElements(i))],node1,node2,node3,node4,rcWall_AreaConf_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements(i)});
end % for i

% We add the Center Elements
numberElements = length(centerElements);
for i = 1:numberElements
    node1 = elements2D(centerElements(i),2);
    node2 = elements2D(centerElements(i),3);
    node3 = elements2D(centerElements(i),4);
    node4 = elements2D(centerElements(i),5);
    quad4NObjects {centerElements(i)} = Quad4NR02(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {centerElements(i)});
end % for i

else
% We add a parameter to the Name of the File to save
nameFileMat = [ nameFileMat " ];
% Elements without Drilling DOF
% We add the Foundation Elements
numberElements_Area = length(foundationElements);
for i = 1:numberElements_Area
    node1 = elements2D(foundationElements(i),2);
    node2 = elements2D(foundationElements(i),3);
    node3 = elements2D(foundationElements(i),4);
    node4 = elements2D(foundationElements(i),5);
    quad4NObjects {foundationElements(i)} = Quad4N(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {foundationElements(i)});
end % for i

% We add the Foundation Elements
    numberElements_Area = length(superiorElements);
for i = 1:numberElements_Area

```



```

    node1 = elements2D(superiorElements(i),2);
    node2 = elements2D(superiorElements(i),3);
    node3 = elements2D(superiorElements(i),4);
    node4 = elements2D(superiorElements(i),5);
    quad4NObjects {superiorElements(i)} = Quad4N(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {superiorElements(i)});
end % for i

% We add the Boundary Elements
numberElements_Area = length(boundaryElements);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements(i),2);
    node2 = elements2D(boundaryElements(i),3);
    node3 = elements2D(boundaryElements(i),4);
    node4 = elements2D(boundaryElements(i),5);
    quad4NObjects {boundaryElements(i)} = Quad4N(['E'
num2str(boundaryElements(i))],node1,node2,node3,node4,rcWall_AreaConf_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements(i)});
end % for i

% We add the Center Elements
numberElements = length(centerElements);
for i = 1:numberElements
    node1 = elements2D(centerElements(i),2);
    node2 = elements2D(centerElements(i),3);
    node3 = elements2D(centerElements(i),4);
    node4 = elements2D(centerElements(i),5);
    quad4NObjects {centerElements(i)} = Quad4N(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {centerElements(i)});
end % for i

end % if optionTypeElement
numberElements = size(elements2D,1);

%% Create the Constraint
numberConstraints = length(nodesSPC);
spcObjects = cell(numberConstraints,1);
for i=1:numberConstraints
    spcObjects {i} = FixRestraint(['SPC_N' num2str(nodesSPC(i))],nodesSPC(i),[1 2 3]);
    % We add the SP_Constraint
    modelObj.addSP_Constraint(spcObjects {i});
end % for i

%% We store the time to take create the Model

```

```
timeModel = toc;
fprintf('Time to initialize the Model Object and ModelComponents Objects is %s [sec]
\n',num2str(timeModel));
```

```
%% We proceed to set The type of Nodes Reactions to Calculate
typeReactions = 0;
flagNodesReactions = true;
modelObj.setTypeNodesReactions(typeReactions,flagNodesReactions);
```

Generador de Mallado

```
Mesh_fc2(2,1220,3660,9,16,95,96,305,'Nodes_RW2_RW.txt','Elements_RW2_RW.txt')
```

Desplazamiento por Flexión y Corte

```
%% Opcion 1: Promedio por piso
```

```
%Obtener el desplazamiento angular para el primer piso
```

```
boundaryElements_1st = [57 : 58 65 : 66 67 : 68
75 : 76 77 : 78 85 : 86 87 :
88 95 : 96 97 : 98 105 : 106];
centerElements_1st = [59 : 64 69 : 74
79 : 84 89 :
94 99 : 104];
```

```
%Cantidad de Elementos de borde
```

```
nBoundary=2;
```

```
%Numero de Elementos por piso
```

```
if nBoundary==1
```

```
    nHorizontal=boundaryElements_1st(2)-boundaryElements_1st(1)+1;
```

```
elseif nBoundary==2
```

```
    nHorizontal=boundaryElements_1st(4)-boundaryElements_1st(1)+1;
```

```
end
```

```
totalElements_Wall=length(boundaryElements_1st)+length(centerElements_1st);%Cantidad de
Elementos Verticales
```

```
%Matrices iniciales
```

```
angularElements_Result=zeros(numberTotalIncrRecorded_SHL,totalElements_Wall);
```

```
disp_ang_Element=zeros(numberTotalIncrRecorded_SHL,totalElements_Wall);
```

```
elementHigh=zeros(totalElements_Wall,1);
```

```
%Contadores
```

```
elementCounter=boundaryElements_1st(1);
```

```
element_b=1;
```

```
element_c=1;
```

```

for i=1:totalElements_Wall
    %Respuesta de todos los puntos de cuadratura (9)

response=shlElementRRObjets {elementCounter}.getResponseRecorded('StrainAtQuadraturePoints');
    %Promedio de los puntos de cuadratura

meanResponse=(response(:,3)+response(:,6)+response(:,9)+response(:,12)+response(:,15)+response(:,18)+response(:,21)+response(:,24)+response(:,27))/9;
    angularElements_Result(:,i)=meanResponse;

    %Altura del elemento
    n1=elements2D(elementCounter,2);
    n2=elements2D(elementCounter,5);
    elementHigh(i)=nodes2D(n2,3)-nodes2D(n1,3);

    %Deflexion del elemento
    disp_ang_Element(:,i)=meanResponse*elementHigh(i);

    %Suma a contadores
    elementCounter=elementCounter+1;
    if elementCounter==boundaryElements_1st(element_b)
        element_b=element_b+1;
    else
        element_c=element_c+1;
    end
end

%Deflección por piso
dispPerFloor=zeros(numberTotalIncrRecorded_SHL,totalElements_Wall/nHorizontal);
inHorizontal=1;
fnHorizontal=inHorizontal+(nHorizontal-1);
for i=1:(totalElements_Wall/nHorizontal)
    for j=inHorizontal:fnHorizontal
        dispPerFloor(:,i)=dispPerFloor(:,i)+disp_ang_Element(:,j);
    end
    inHorizontal=inHorizontal+nHorizontal;
    fnHorizontal=fnHorizontal+nHorizontal;
    dispPerFloor(:,i)=dispPerFloor(:,i)/nHorizontal;
end

%deflección total por corte
Wall_displ_Prog(:,2)=(sum(dispPerFloor'))';

%deflección por momento
Wall_displ_Prog(:,3)=Wall_displ_Prog(:,1)-Wall_displ_Prog(:,2);
%figure

```

```

%plot(Wall_displ_Opt1(:,1),Wall_load,Wall_displ_Opt1(:,2),Wall_load,Wall_displ_Opt1(:,3),Wall_lo
ad);legend('total','corte','momento')

%% Opcion 3: Usando movimiento de los puntos

l=abs(nodes2D(nodes_shearDisp(1),2)-nodes2D(nodes_shearDisp(4),2));
h=abs(nodes2D(nodes_shearDisp(1),3)-nodes2D(nodes_shearDisp(4),3));

disp1st_n2=shlNodeRRObjects {nodes_shearDisp(4)}.getResponseRecorded('Displ');
disp1st_n1=shlNodeRRObjects {nodes_shearDisp(3)}.getResponseRecorded('Displ');

U1=disp1st_n1(:,1);
V1=disp1st_n1(:,2);

U2=disp1st_n2(:,1);
V2=disp1st_n2(:,2);

d1=sqrt(h^2+l^2);
d2=d1;

d1_d=sqrt((h+V2).^2+(l+U2).^2);
d2_d=sqrt((h+V1).^2+(l-U1).^2);

gamma_avg=((d1_d-d1).*d1-(d2_d-d2).*d2)/(2*h*l);

Wall_displ_Thom(:,2)=gamma_avg*h;

%deflección por momento
Wall_displ_Thom(:,3)=Wall_displ_Thom(:,1)-Wall_displ_Thom(:,2);

%% Grafico de Figuras

Exp_Results_Corte=load('RW2_ExperimentalResults_1st_Shear.txt');
figure
plot(Exp_Results_Corte(:,2),Exp_Results_Corte(:,1),Wall_displ_Prog(:,2),Wall_load,Wall_displ_Tho
m(:,2),Wall_load); legend('Exp','Prog','Thom')
xlabel('Comparacion de calculo de desplazamiento por corte [mm]');ylabel('Fuerza [kN]');
grid on

Exp_Results_Flex=load('RW2_ExperimentalResults_1st_Flex.txt');
figure
plot(Exp_Results_Flex(:,2),Exp_Results_Flex(:,1),Wall_displ_Prog(:,3),Wall_load,Wall_displ_Thom(
:,3),Wall_load); legend('Exp','Prog','Thom')
xlabel('Comparacion de calculo de desplazamiento por flexion [mm]');ylabel('Fuerza [kN]');
grid on

```

Gráfico de Mallado

```

%% Define Variables Depending of the Model
NameAxesModel = {'Modelaci?n de Muro - MR2'};

%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

% We create the AxesModelHandler

a=figure();
set(a,'Position',[ 1025 545 1100*1.3 850*1.3 ],'papersize',[11 8.5],'paperposition',[0 0 11
8.5],'color','w');
axesModelHandlerObjects {1} = AxesModelHandler(NameAxesModel {1},a);%,[.05 .05 .4 .4]);

% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {1});
setBackgroundcolor(axesModelHandlerObjects {1},'white')
% We set the LoadPattern to Draw in the AxesModel 1
shlModelGraphicHandlerObj.setLoadPatternToDraw(NameAxesModel {1},shlLoadPatternTag,'Undeformed');
shlModelGraphicHandlerObj.drawElements(axesModelHandlerObjects {1});
shlModelGraphicHandlerObj.drawNodes(axesModelHandlerObjects {1});
%shlModelGraphicHandlerObj.drawNodeTags(axesModelHandlerObjects {1});
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {1},sizeSimbolSPC);
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementShadow')
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementClass')
%title('Display');

timePerformAnalysis = toc;
fprintf("Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));
%title('Modelo Muro MR1');

```

Gráfico de Tensión y Deformación

```

%% Define Variables Depending of the Model
NameAxesModel
= {'RW2_TW_Undeformed','RW2_TW_Deformed','RW2_TW_Undeformed_CrackPattern'};
newGraphicScaleFactor = 1000;
sizeSimbolSPC = 40;

tic
%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

% We create the AxesModelHandler

```

```

Gs = cell(1);
axesModelHandlerObjects = cell(6,1);

Gs{1} = figure();
set(Gs{1},'Units','inches','Position',[ 10 6 8.5 11 ],'papersize',[8.5 11],'paperposition',[0 0 8.5
11],'color','w','defaultFigurePaperPositionMode','auto');

tic

axesModelHandlerObjects{1} = AxesModelHandler('RW2_TW_Undeformed_S11',Gs{1},[.05 .5
.3 0.5]);
axesModelHandlerObjects{2} = AxesModelHandler('RW2_TW_Undeformed_S22',Gs{1},[.35 .5
.3 0.5]);
axesModelHandlerObjects{3} = AxesModelHandler('RW2_TW_Undeformed_S12',Gs{1},[.65 .5
.3 0.5]);
axesModelHandlerObjects{4} = AxesModelHandler('RW2_TW_Undeformed_E11',Gs{1},[.05 .0
.3 .5]);
axesModelHandlerObjects{5} = AxesModelHandler('RW2_TW_Undeformed_E22',Gs{1},[.35 .0
.3 .5]);
axesModelHandlerObjects{6} = AxesModelHandler('RW2_TW_Undeformed_E12',Gs{1},[.65 .0
.3 .5]);

setBackgroundColor(axesModelHandlerObjects{1},'white')
setBackgroundColor(axesModelHandlerObjects{2},'white')
setBackgroundColor(axesModelHandlerObjects{3},'white')
setBackgroundColor(axesModelHandlerObjects{4},'white')
setBackgroundColor(axesModelHandlerObjects{5},'white')
setBackgroundColor(axesModelHandlerObjects{6},'white')

% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{1});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{2});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{3});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{4});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{5});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{6});

% We set the type of ResponseToPlot in each AxesModel
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S11','S11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S22','S22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S12','S12','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E11','E11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E22','E22','Undeformed');

```

```
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E12','E12','Undeformed');
```

```
% We set the Scale of the Deformation to draw
```

```
shlModelGraphicHandlerObj.setGraphicDeformationScaleFactor(newGraphicScaleFactor);
```

```
% We proceed to draw the Last Number of AnalysisStep
```

```
for i= Step2Draw
```

```
    % First we set the AnalysisStep to Draw
```

```
    shlModelGraphicHandlerObj.setAnalysisStepNumber(i);
```

```
    % We draw the Elemets
```

```
shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {1},true);
```

```
shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {2},true);
```

```
shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {3},true);
```

```
shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {4},true);
```

```
shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {5},true);
```

```
shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {6},true);
```

```
    % We draw the SP_Constraint
```

```
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {1},sizeSimbolSPC);
```

```
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {2},sizeSimbolSPC);
```

```
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {3},sizeSimbolSPC);
```

```
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {4},sizeSimbolSPC);
```

```
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {5},sizeSimbolSPC);
```

```
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {6},sizeSimbolSPC);
```

```
end
```

```
timePerformAnalysis = toc;
```

```
fprintf("Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));
```

Ejecutor

```
%We close all the Windows
```

```
close all
```

```
opt = 0;
```

```
if opt==0
```

```

% We clear the memory and the classes and functions
clear classes;
clear all;
elseif opt == 1
% We clear the model, analysis model and modelGraphicHandler
modelObj.clearAll();
svlModelResponseRecorderObj.clearAll()
svlAnalysisModelObj.clearAll()
shlModelResponseRecorderObj.clearAll()
shlAnalysisModelObj.clearAll()
shlModelGraphicHandlerObj.clearAll();
clear variables
clear all;
elseif opt == 2
modelObj.clearAll();
svlModelResponseRecorderObj.clearAll()
shlModelResponseRecorderObj.clearAll()
shlModelGraphicHandlerObj.clearAll();
clear variables
clear all;
end
clc

nameFileMat = 'Output_RW2_RW2';

% Option to create the Model

% Save or Load Option:
%          1 = Save
%          2 = Load
optionSaveOrLoad = 1;

% Mesh:
%          1 = Simple Mesh (170mm)
optionMesh = 1;

% Specimen Wall Option:
%          2 = RW2
optionSpecimenWall = 2;

% Concrete Option:
%          1 =
%          OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01
optionConcrete = 1;

% Steel Option:
%          1 = SmearredEmbeddedSteelDoubleLayerROT2DMaterial01
%          2 = SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01
optionSteel = 3;

```



```

% Damage Option:
%          0 = No Damage
%          1 = Damage
optionDamage = 1;

% Type of Element:
%          1 = Quad4N
%          2 = Quad4NR02
optionTypeElement = 2;

% SelfWeight Option:
%          0 = Not Include
%          1 = Include
optionSelfWeight = 1;

%%% We create the Model
RW2_RW_Model

%%% We Create the Static Vertical Load, the Response Recorder and Perform the Analysis
RW2_RW_StaticVerticalLoad

%%% We Create the Static Horizontal Load, the Response Recorder and Perform the Analysis
RW2_RW_StaticHorizontalLoad

%%% We create the Graphic for the Static Horizontal Load
iNumberIncrementRecorded_SHL = numberTotalIncrRecorded_SHL-1;

% We extract the Displacement of the Top of the Wall
Wall_disp =
shlNodeRRObjets {nodeTagToMonitorInTheDisplacementControl}.getResponseRecorded('Displ');
% We extract the Total Horizontal Load apply at the Top of the Wall
loadCM = numberHorizontalNodeLoads*shlNodeLoadRRObjets {1}.getResponseRecorded();
%%% We Create Variable of Displacement
Wall_displ_Prog=zeros(numberTotalIncrRecorded_SHL,3);
disp1st=shlNodeRRObjets {120}.getResponseRecorded('Displ');

Wall_displ_Prog(:,1)=disp1st(:,1);
Wall_load=loadCM(:,dofNumberHorizontalLoad {1})/1000;

Wall_displ_Thom=zeros(numberTotalIncrRecorded_SHL,3);
Wall_displ_Thom(:,1)=disp1st(:,1);

%%% Graph Step 1%
one=find(Wall_disp(:,1) > height_Wall/100 & Wall_disp(:,1) < height_Wall/100+1);
Step2Draw = one(1);
RW2_RW_GraphicInStep

```

```

%% Graph Step Max V
Step2Draw=find(Wall_load==max(Wall_load));
RW2_RW_GraphicInStep

%% Graphic Element
RW2_RW_ElementGraphic

%% Shear and Moment Graphic
RW2_ShearFlexion

%%
% We create the Figure that compare the top wall displacement response between the Experiment vs
the Analytical response
fg = figure();
% We plot the graphics
Exp_Results=load('RW2_ExperimentalResults.txt');
plot(Exp_Results(:,1),Exp_Results(:,2),'b','LineWidth',0.5);
hold on
plot(Wall_disp(:,1),Wall_load,'Color',[1 0 0],'LineWidth',0.5);
xlabel('Desplazamiento [mm]');
ylabel('Fuerza [kN]');
legend('RW2','MR1');
xlim([-100,100])
grid on
savefig('FiguraAux_RW2');

```

Código MR2

Modelo

```

tic
%% Create the Model
modelObj=Model();

nodes2D = load('Nodes_S78_RW.txt');
numberNodes = length(nodes2D);
elements2D = load('Elements_S78_RW.txt');
numberElements = size(elements2D,1);
nodesSPC = 1 : 13;
horizontalNodeLoads = [147 156 165];
verticalNodeLoads = 175:181;
nodeTagToMonitorInTheDisplacementControl = 156;%138;
nodes_shearDisp= [55 63 138 146];
foundationElements = 1 : 48;
superiorElements= 121 : 152;
boundaryElements = [49 : 50 55 : 56 57 : 58
63 : 64 65 : 66 71 : 72 73 :
74 79 : 80 81 : 82 87 : 88
89 : 90 95 : 96 97 : 98 103
: 104 105 : 106 111 : 112 113 : 114
119 : 120];
centerElements = [51 : 54 59 : 62
67 : 70 75 : 78
83 : 86 91 :
94 99 : 102
107 : 110 115 : 118];

%% Create the Nodes
if optionTypeElement == 1
    numDOF = 2;
else
    numDOF = 3;
end % if

nodesObjects = cell(numberNodes,1);
for i = 1:numberNodes
    nodesObjects {i} = Node(i,numDOF,nodes2D(i,2:3));
    % Add the Nodes that define the element
    modelObj.addNode(nodesObjects {i});
end % for i

%% Material Properties
magnitudGSelfWeightLoad = 9800; % [mm/s^2]
rhoConcreteMaterial = 24.5*(10^-6)/magnitudGSelfWeightLoad; % [N/mm^3] Concrete
nuConcrete = 0.2;

```

```

% Steel
%Be Steel - "Confined"
Es_BeVertical = 200000; % [MPa]
Fy_BeVertical = 477; % [MPa]
strainAtFy_BeVertical = Fy_BeVertical/Es_BeVertical; % [mm/mm]
d_BarBeVertical = (19.05+15.8)/2; % [mm]
Area_BeVertical = pi*(d_BarBeVertical/2)^2; % [mm^2]

%Vertical Steel
Es_Vertical = 200000; % [MPa]
Fy_Vertical = 443; % [MPa]
strainAtFy_Vertical = Fy_Vertical/Es_Vertical; % [mm/mm]
d_BarVertical = 9.52; % [mm]
Area_Vertical = pi*(d_BarVertical/2)^2; % [mm^2]

%Horizontal Steel
Es_Horizontal = 200000; % [MPa]
Fy_Horizontal = 443; % [MPa]
strainAtFy_Horizontal = Fy_Horizontal/Es_Horizontal; % [mm/mm]
d_BarHorizontal = 9.52; % [mm]
Area_Horizontal = pi*(d_BarHorizontal/2)^2; % [mm^2]

%Hoops Steel
Es_Hoops = 200000; % [MPa]
Fy_Hoops = 423; % [MPa]
strainAtFy_Hoops = Fy_Hoops/Es_Hoops; % [mm/mm]
d_BarHoops = 6.35; % [mm]
Area_Hoops = pi*(d_BarHoops/2)^2; % [mm^2]

% We add a parameter to the Name of the File to save
nameFileMat = [ nameFileMat '_WR' num2str(optionSpecimenWall) ];

%% Geometry of Wall
% Dimension Rectangular Wall

height_Wall = 81*25.4; % [mm]
thickness_Wall = 152; % [mm]
coverConcrete_Wall = 1.125*25.4-19.05/2-d_BarHoops; % [mm]
lengthBoundary_Wall = 9*25.4; % [mm]
lengthCenter_Wall = 48*25.4-2*lengthBoundary_Wall; % [mm]

% We Set the Total Horizontal Force use to Analysis with Displacement
% Control
totalHorizontalForceLoad = 10000; %[N]

% We define the Total Vertical Load
totalVerticalForceLoad = -641000; %[N]

```

```

%% %%DEFINIDO POR ENSAYOS
% We define the Lateral Drift Procedure
lateralDrift = [0 0.1 -0.1 0.2 -0.2 0.3 -0.3 0.6 -0.5 0.57 -0.51 0.61 -0.5 1.2 -1.2 1.3 -1.2 1.3 -1.2 1.3
    3.8 -3.4 3.4 -3.5 3.8 -3.5 6 -6.7 7.0 -6.9 7.0 -6.9 6.9
    -6.9 9.3 -9.2 9.3 -9.2 9.3 -9.1 13.9 -13.7 13.8 -13.7 14.0
    -13.7 18.6 -18.2 18.5 -18.2 18.6 -18.2 27.8 -27.3 28.1 -27.3 27.8
    -27.3 29.5 36.9 -36.2 37.0 -36.4 36.9 0 -36.4 0:55 55.2 55:-1:-54 -54.7
    55.4 -54.7 0];%ciclo
%% %%DEFINIDO POR ENSAYOS
%% %%

numberLateralDrift = length(lateralDrift);
time = zeros(numberLateralDrift,1);
numberStepsPerCicle = zeros(numberLateralDrift,1);
totalTime = 0;
slope = 1/1;
for i=1:numberLateralDrift
    lateralDrift(i) = sign(lateralDrift(i))*ceil(abs(lateralDrift(i))*10)/10;
end % for i
for i=1:numberLateralDrift-1
    totalTime = totalTime + ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
    time(i+1) = totalTime;
    numberStepsPerCicle(i+1) = ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
end % for i
fprintf('Number of Steps to Perform is: %s \n',num2str(totalTime));

factorReductionFc = 1;

%% %% Confinamiento

% Concrete
Fc_Pedestal = -45; % [MPa]
Ec_Pedestal = 4700*sqrt(-Fc_Pedestal); % [MPa]

Fc = -55.8*factorReductionFc; % [MPa]
strainAtFc_Center = -0.003; % [mm/mm]
strainAtFc_Boundary = -0.0055; % [mm/mm]
Fcr = 0.31*sqrt(-Fc); % [MPa]
strainAtFcr = 0.00008; % [mm/mm]

%% %% Cuantias de Armaduras

rho_Hoops_Area = 0.0083;
rho_BeVerticalSteel_Area = 0.0615;
rho_HorizontalSteel_Area = 0.0063;
rho_VerticalSteel_Area = 0.0071;

```

```

%% Steel Constant

ratioRhoFyoverFc = abs(rho_Hoops_Area*Fy_Hoops/Fc);
Kconf_Opt1 = 1 + ratioRhoFyoverFc; % Orakcal et al. 2006
Kconf_Opt2 = 1 + 0.92*ratioRhoFyoverFc - 0.76*ratioRhoFyoverFc^2; % Vechio 1992 ( Kupfer et
al 1969 )
Kconf_Opt3 = -1.254 + 2.254*sqrt(1+7.94*(ratioRhoFyoverFc)) - 2*(ratioRhoFyoverFc); % Mander
et al (1988)
Kconf = Kconf_Opt1; % Use

%% Create the 2D Material
% Create the Smearred Steel 2D Material
% Steel 2D Material Class Parameters:

if optionSteel == 1
    nameFileMat = [ nameFileMat '_SES' ];
    bsh = 0.01;
    Ac = 1.9;
    Rc = 10;
    OrientationEmbeddedSteel = 0;

    smearedESteel_Area_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerROT2DMaterial01('SmearredESteel_Area',[Es_Horizontal,Es_
    Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[2*rho_HorizontalSteel_Area,2*rho_VerticalSteel_Ar
    ea],OrientationEmbeddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Center,0);
    smearedESteel_AreaBe_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerROT2DMaterial01('SmearredESteel_AreaBe',[Es_Horizontal,Es
    _BeVertical],[(Fy_Horizontal*rho_HorizontalSteel_Area +
    Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Fy_BeVertical],[Fcr,F
    cr],[2*(rho_HorizontalSteel_Area+rho_Hoops_Area),2*rho_BeVerticalSteel_Area],OrientationEmb
    eddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Boundary,0);

elseif optionSteel == 2
    nameFileMat = [ nameFileMat '_SS' ];
    OrientationEmbeddedSteel = 0;
    R0 = 18;
    CR1 = 0.9;
    CR2 = 0.15;
    a1 = -0.002; %To model Damage in the steel
    a2 = 1;
    a3 = -0.002; %To model Damage in the steel
    a4 = 1;
    smearedESteel_Area_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearredESteel_Area',[Es_Horizontal,Es_
    Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_
    Area,rho_VerticalSteel_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,str
    ainAtFc_Center,0);
    smearedESteel_AreaBe_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearredESteel_AreaBe',[Es_Horizontal,Es

```

```
_BeVertical],[Fy_Horizontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Fy_BeVertical],[Fcr,F
cr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_Area+rho_Hoops_Area,rho_BeVerticalStee
l_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,strainAtFc_Boundary,0);
```

```
elseif optionSteel == 3
```

```
nameFileMat = [ nameFileMat '_SSP' ];
OrientationEmbeddedSteel = 0;
Bsh=0.01;
R0 = 18;
CR1 = 0.9;
CR2 = 0.15;
a1 = -0.002; %To model Damage in the steel
a2 = 1;
a3 = -0.002; %To model Damage in the steel
a4 = 1;
```

```
M=38;
Ecase = 1/300;
Ehtens = 0.04;
Eutens = 0.11;
Sigutens = 630;
Eohtens = 7000;
Ef=0.55;
Ff=0;
InitialStress = 0;
```

```
steel1UniaxialLayer_Area=SteelUniaxialMaterial('SteelUniaxial_Area',Es_Vertical,Fy_Vertical,Bsh,R0
,CR1,CR2,a1,a2,a3,a4);
```

```
steel2UniaxialLayer_Area=SteelUniaxialMaterial('SteelMMUniaxial_Area',Es_Horizontal,Fy_Horizo
ntal,Bsh,R0,CR1,CR2,a1,a2,a3,a4);
```

```
steel1UniaxialLayer_AreaBe=SteelUniaxialMaterial('SteelUniaxial_AreaBe',Es_BeVertical,(Fy_Horiz
ontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Bsh,R0,CR1,CR2,a1,a
2,a3,a4);
```

```
steel2UniaxialLayer_AreaBe=SteelMMUniaxialMaterial('SteelMMUniaxial_AreaBe',M,d_BarBeVertic
al,76,Ecase,440,Es_BeVertical,Ehtens,Eutens,Sigutens,Eohtens,Ef,Ff,InitialStress);
```

```
smearESteel_Area_2DMatObj =
SteelDoubleLayer2DMaterial('SmearESteel_Area',rho_HorizontalSteel_Area,steel1UniaxialLayer_Ar
ea,rho_VerticalSteel_Area,steel2UniaxialLayer_Area,OrientationEmbeddedSteel);
```

```

    smearedESteel_AreaBe_2DMatObj =
    SteelDoubleLayer2DMaterial('SmearedSteel_AreaBe',rho_HorizontalSteel_Area+rho_Hoops_Area,s
    teel1UniaxialLayer_AreaBe,rho_BeVerticalSteel_Area,steel2UniaxialLayer_AreaBe,OrientationEmbe
    ddedSteel);

end

%%% Create the Concrete Uniaxial Material
if optionDamage == 0
    damageConstantUnconf_1 = 0;
    damageConstantUnconf_2 = 0;
    damageConstantConf_1 = 0;
    damageConstantConf_2 = 0;
elseif optionDamage == 1
    damageConstantUnconf_1 = 0.1;
    damageConstantUnconf_2 = 0.4;
    damageConstantConf_1 = 0.1;
    damageConstantConf_2 = 0.4;
end % if optionConcrete

%%% Create the Orthotropic Rotating Angle 2D Material

% Tension Stiffening Concrete 2DMaterial
nameFileMat = [ nameFileMat '_TS' ];
K = 1;
AlphaC = 0.32;
AlphaT = 0.08;

% Unconfined
R = 3.1;
B = 0.6;
orthotropicConcrete_Area_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_Area',Fc
,strainAtFc_Center,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstantUnconf_1,da
mageConstantUnconf_2,rhoConcreteMaterial);

% Confined
R = 2.1;
B = 0.4;
orthotropicConcrete_AreaBe_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_AreaCo'
,Kconf*Fc,strainAtFc_Boundary,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstan
tConf_1,damageConstantConf_2,rhoConcreteMaterial);

if optionDamage == 0
elseif optionDamage == 1
    % We add a parameter to the Name of the File to save

```



```

    nameFileMat = [ nameFileMat 'wUD_1_5' ];
end % if optionConcrete

if optionSelfWeight == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_SW' ];
end % if

%% Profile Properties
% We create the profile for the Reinforced Concrete Wall
coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_Area_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_Area',coordinatesConcreteLayers,c
oordinatesReinforcedSteelLayers,thickness_Wall);

coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_AreaConf_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_AreaBe',coordinatesConcreteLayer
s,coordinatesReinforcedSteelLayers,thickness_Wall);

%Create the Reinforced Concrete Layered Membrane Section
concrete2DMaterialObjectAtEachLayer = {orthotropicConcrete_Area_RA2DMatObj};
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_Area_2DMatObj };
rcWall_Area_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_Area',rcWall_Area_ProfileObj,concr
ete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

concrete2DMaterialObjectAtEachLayer = { orthotropicConcrete_AreaBe_RA2DMatObj };
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_AreaBe_2DMatObj };
rcWall_AreaConf_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_AreaConf',rcWall_AreaConf_Profile
Obj,concrete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

%Create the Elastic Concrete Membrane Pedestal Section

% Create the 2D Material
elasticConcrete2DMat =
ElasticIsotropicPlaneStress2DMaterial('ElasticConcrete2DMat',Ec_Pedestal,nuConcrete,rhoConcret
eMaterial);
thickness_Pedestal = 915; % [mm]
rcWall_Pedestal_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_Pedestal);
rcWall_Pedestal_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_Pedestal_ProfileObj,elasticConcrete2DMat);

%Create the Elastic Concrete Membrane Superior Section
%Create the 2D Material

```

```

thickness_SupSection = thickness_Wall; % [mm]
rcWall_SupSection_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_SupSection);
rcWall_SupSection_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_SupSection_ProfileObj,elasticConcrete2DMat);

%% Create the Element
quad4NObjects = cell(numberElements,1);
if optionTypeElement == 2
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_Rot' ];
    % Elements with Drilling DOF

    % We add the Foundation Elements
    numberElements_Area = length(foundationElements);
    for i = 1:numberElements_Area
        node1 = elements2D(foundationElements(i),2);
        node2 = elements2D(foundationElements(i),3);
        node3 = elements2D(foundationElements(i),4);
        node4 = elements2D(foundationElements(i),5);
        quad4NObjects { foundationElements(i) } = Quad4NR02(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj,optionTypeElement);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects { foundationElements(i) });
    end % for i

    % We add the Superior Elements
    numberElements_Area = length(superiorElements);
    for i = 1:numberElements_Area
        node1 = elements2D(superiorElements(i),2);
        node2 = elements2D(superiorElements(i),3);
        node3 = elements2D(superiorElements(i),4);
        node4 = elements2D(superiorElements(i),5);
        quad4NObjects { superiorElements(i) } = Quad4NR02(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj,optionTypeElement);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects { superiorElements(i) });
    end % for i

    % We add the Boundary Elements
    numberElements_Area = length(boundaryElements);
    for i = 1:numberElements_Area
        node1 = elements2D(boundaryElements(i),2);
        node2 = elements2D(boundaryElements(i),3);
        node3 = elements2D(boundaryElements(i),4);

```

```

        node4 = elements2D(boundaryElements(i),5);
        quad4NObjects {boundaryElements(i)} = Quad4NR02(['E'
num2str(boundaryElements(i))],node1,node2,node3,node4,rcWall_AreaConf_SectionObj,optionTyp
eElement);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {boundaryElements(i)});
    end % for i

% We add the Center Elements
numberElements = length(centerElements);
for i = 1:numberElements
    node1 = elements2D(centerElements(i),2);
    node2 = elements2D(centerElements(i),3);
    node3 = elements2D(centerElements(i),4);
    node4 = elements2D(centerElements(i),5);
    quad4NObjects {centerElements(i)} = Quad4NR02(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj,optionTypeElemen
t);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {centerElements(i)});
end % for i

else
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat " "];
    % Elements without Drilling DOF
    % We add the Foundation Elements
    numberElements_Area = length(foundationElements);
    for i = 1:numberElements_Area
        node1 = elements2D(foundationElements(i),2);
        node2 = elements2D(foundationElements(i),3);
        node3 = elements2D(foundationElements(i),4);
        node4 = elements2D(foundationElements(i),5);
        quad4NObjects {foundationElements(i)} = Quad4N(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {foundationElements(i)});
    end % for i

% We add the Foundation Elements
    numberElements_Area = length(superiorElements);
    for i = 1:numberElements_Area
        node1 = elements2D(superiorElements(i),2);
        node2 = elements2D(superiorElements(i),3);
        node3 = elements2D(superiorElements(i),4);
        node4 = elements2D(superiorElements(i),5);
        quad4NObjects {superiorElements(i)} = Quad4N(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj);
        % Add the Elements that define the element

```

```

        modelObj.addElement(quad4NObjects {superiorElements(i)});
    end % for i

    % We add the Boundary Elements
    numberElements_Area = length(boundaryElements);
    for i = 1:numberElements_Area
        node1 = elements2D(boundaryElements(i),2);
        node2 = elements2D(boundaryElements(i),3);
        node3 = elements2D(boundaryElements(i),4);
        node4 = elements2D(boundaryElements(i),5);
        quad4NObjects {boundaryElements(i)} = Quad4N(['E'
num2str(boundaryElements(i))],node1,node2,node3,node4,rcWall_AreaConf_SectionObj);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {boundaryElements(i)});
    end % for i

    % We add the Center Elements
    numberElements = length(centerElements);
    for i = 1:numberElements
        node1 = elements2D(centerElements(i),2);
        node2 = elements2D(centerElements(i),3);
        node3 = elements2D(centerElements(i),4);
        node4 = elements2D(centerElements(i),5);
        quad4NObjects {centerElements(i)} = Quad4N(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {centerElements(i)});
    end % for i

end % if optionTypeElement
numberElements = size(elements2D,1);

%% Create the Constraint
numberConstraints = length(nodesSPC);
spcObjects = cell(numberConstraints,1);
for i=1:numberConstraints
    spcObjects {i} = FixRestraint(['SPC_N' num2str(nodesSPC(i))],nodesSPC(i),[1 2 3]);
    % We add the SP_Constraint
    modelObj.addSP_Constraint(spcObjects {i});
end % for i

%% We store the time to take create the Model
timeModel = toc;
fprintf("Time to initialize the Model Object and ModelComponents Objects is %s [sec]
\n',num2str(timeModel));

%% We proceed to set The type of Nodes Reactions to Calculate
typeReactions = 0;

```

```
flagNodesReactions = true;
modelObj.setTypeNodesReactions(typeReactions,flagNodesReactions);
```

Generador de Mallado

```
Mesh_fc2(2,1220,1828,7,8,100,110,228,'Nodes_S78_RW.txt','Elements_S78_RW.txt')
```

Desplazamiento por Flexión y Corte

```
%% Opcion 1: Promedio por piso
```

```
Wall_displ_Prog(:,1)=Wall_displ_Thom(:,1);
%Obtener el desplazamiento angular para el primer piso
```

```
boundaryElements_Prog = [49      :      50      55      :      56      57      :
      58      63      :      64      65      :      66      71      :      72
      73      :      74      79      :      80      81      :      82      87
      :      88      89      :      90      95      :      96      97      :      98
      103      :      104      105      :      106      111      :      112      113      :
      114      119      :      120];
centerElements_Prog = [51      :      54      59      :      62
      67      :      70      75      :
      78      83      :      86
      91      :      94      99      :      102
      107      :      110      115      :      118];
```

```
%Cantidad de Elementos de borde
```

```
nBoundary=2;
```

```
%Numero de Elementos por piso
```

```
if nBoundary==1
```

```
    nHorizontal=boundaryElements_Prog(2)-boundaryElements_Prog(1)+1;
```

```
elseif nBoundary==2
```

```
    nHorizontal=boundaryElements_Prog(4)-boundaryElements_Prog(1)+1;
```

```
end
```

```
totalElements_Wall=length(boundaryElements_Prog)+length(centerElements_Prog+1);%Cantidad
de Elementos Verticales
```

```
%Matrices iniciales
```

```
angularElements_Result=zeros(numberTotalIncrRecorded_SHL,totalElements_Wall);
```

```
disp_ang_Element=zeros(numberTotalIncrRecorded_SHL,totalElements_Wall);
```

```
elementHigh=zeros(totalElements_Wall,1);
```

```
%Contadores
```

```
elementCounter=boundaryElements_Prog(1);
```

```
element_b=1;
```

```
element_c=1;
```

```

for i=1:totalElements_Wall
    %Respuesta de todos los puntos de cuadratura (9)

response=shlElementRRObjets {elementCounter}.getResponseRecorded('StrainAtQuadraturePoints');
    %Promedio de los puntos de cuadratura

meanResponse=(response(:,3)+response(:,6)+response(:,9)+response(:,12)+response(:,15)+response(:,18)+response(:,21)+response(:,24)+response(:,27))/9;
    angularElements_Result(:,i)=meanResponse;

    %Altura del elemento
    n1=elements2D(elementCounter,2);
    n2=elements2D(elementCounter,5);
    elementHigh(i)=nodes2D(n2,3)-nodes2D(n1,3);

    %Deflexion del elemento
    disp_ang_Element(:,i)=meanResponse*elementHigh(i);

    %Suma a contadores
    elementCounter=elementCounter+1;
    if elementCounter==boundaryElements_Prog(element_b)
        element_b=element_b+1;
    else
        element_c=element_c+1;
    end
end

%Deflección por piso
dispPerFloor=zeros(numberTotalIncrRecorded_SHL,totalElements_Wall/nHorizontal);
inHorizontal=1;
fnHorizontal=inHorizontal+(nHorizontal-1);
for i=1:(totalElements_Wall/nHorizontal)
    for j=inHorizontal:fnHorizontal
        dispPerFloor(:,i)=dispPerFloor(:,i)+disp_ang_Element(:,j);
    end
    inHorizontal=inHorizontal+nHorizontal;
    fnHorizontal=fnHorizontal+nHorizontal;
    dispPerFloor(:,i)=dispPerFloor(:,i)/nHorizontal;
end

%deflección total por corte
Wall_displ_Prog(:,3)=(sum(dispPerFloor'))';

%deflección por momento
Wall_displ_Prog(:,2)=Wall_displ_Prog(:,1)-Wall_displ_Prog(:,2);

```

```

%% Obtener desplazamiento producto del corte y de la flexion
% Usando movimiento de los puntos

l=abs(nodes2D(nodes_shearDisp(1),2)-nodes2D(nodes_shearDisp(4),2));
h=abs(nodes2D(nodes_shearDisp(1),3)-nodes2D(nodes_shearDisp(4),3));

dispThom_n2=shlNodeRRObjets {nodes_shearDisp(4)}.getResponseRecorded('Displ');
dispThom_n1=shlNodeRRObjets {nodes_shearDisp(3)}.getResponseRecorded('Displ');

U1=dispThom_n1(:,1);
V1=dispThom_n1(:,2);

U2=dispThom_n2(:,1);
V2=dispThom_n2(:,2);

d1=sqrt(h^2+l^2);
d2=d1;

d1_d=sqrt((h+V2).^2+(l+U2).^2);
d2_d=sqrt((h+V1).^2+(l-U1).^2);

gamma_avg=((d1_d-d1).*d1-(d2_d-d2).*d2)/(2*h*1);

Wall_displ_Thom(:,3)=gamma_avg*h;

%defleccion por momento
Wall_displ_Thom(:,2)=Wall_displ_Thom(:,1)-Wall_displ_Thom(:,3);

%% Opcion 2: Usar el giro
nod_flex=[55 63 84 92 102 110 120 128 138 146];
X_flex=zeros(length(Wall_displ_Thom(:,1)),length(nod_flex));
Y_flex=zeros(length(Wall_displ_Thom(:,1)),length(nod_flex));
for i=1:length(nod_flex)
    disp_nodflex=shlNodeRRObjets {nod_flex(i)}.getResponseRecorded('Displ');
    X_flex(:,i)=disp_nodflex(:,1);
    Y_flex(:,i)=disp_nodflex(:,2);
end

l=abs(nodes2D(nod_flex(1),2)-nodes2D(nod_flex(4),2));
h1=abs(nodes2D(nod_flex(5),3)-nodes2D(nod_flex(7),3));
h2=abs(nodes2D(nod_flex(1),3)-nodes2D(nod_flex(4),3));

SectionA=(0.5)*(Y_flex(:,3)-Y_flex(:,4))/l*h1;
SectionB=(8/15)*(Y_flex(:,5)-Y_flex(:,6))/l*h2;
SectionC=(5/9)*(Y_flex(:,7)-Y_flex(:,8))/l*h2;
SectionD=(2/3)*(Y_flex(:,9)-Y_flex(:,10))/l*h2;

Wall_displ_Tran(:,2)=SectionA+SectionB+SectionC+SectionD;

```

```
Wall_displ_Tran(:,3)=Wall_displ_Tran(:,1)-Wall_displ_Tran(:,2);
```

```
%% Grafico de Figuras
```

```
Exp_Results_Corte=load('S78_ExperimentalResults_Shear.txt');
figure
plot(Exp_Results_Corte(:,2),Exp_Results_Corte(:,1),Wall_displ_Thom(:,3),Wall_load,Wall_displ_Tra
n(:,3),Wall_load,Wall_displ_Prog(:,3),Wall_load); legend('Exp','Thom','Tran')
xlabel('Comparacion de calculo de desplazamiento por corte [mm]');ylabel('Fuerza [kN]');
grid on
```

```
Exp_Results_Flex=load('S78_ExperimentalResults_Flex.txt');
figure
plot(Exp_Results_Flex(:,2),Exp_Results_Flex(:,1),Wall_displ_Thom(:,2),Wall_load,Wall_displ_Tran(
:2),Wall_load,Wall_displ_Prog(:,2),Wall_load); legend('Exp','Thom','Tran')
xlabel('Comparacion de calculo de desplazamiento por flexion [mm]');ylabel('Fuerza [kN]');
grid on
```

Gráfico de Mallado

```
%% Define Variables Depending of the Model
NameAxesModel = {'Modelaci?n de Muro - MR5'};
```

```
%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);
```

```
% We create the AxesModelHandler
```

```
a=figure(22);
set(a,'Position',[ 1025 545 1100*1.3 850*1.3 ],'papersize',[11 8.5],'paperposition',[0 0 11 8.5]);
axesModelHandlerObjects{1} = AxesModelHandler(NameAxesModel{1},a);%,[.05 .05 .4 .4]);
```

```
% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{1});
```

```
% We set the LoadPattern to Draw in the AxesModel 1
shlModelGraphicHandlerObj.setLoadPatternToDraw(NameAxesModel{1},shlLoadPatternTag,'Und
eformed');
shlModelGraphicHandlerObj.drawElements(axesModelHandlerObjects{1});
shlModelGraphicHandlerObj.drawNodes(axesModelHandlerObjects{1});
%shlModelGraphicHandlerObj.drawNodeTags(axesModelHandlerObjects{1});
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects{1},sizeSymbolSPC);
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementShadow')
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementClass')
%title('Display');
```



```
timePerformAnalysis = toc;
fprintf('Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));
```

Gráfico de Tensión y Deformación

```
%% Define Variables Depending of the Model
NameAxesModel
= {'RW2_TW_Undeformed','RW2_TW_Deformed','RW2_TW_Undeformed_CrackPattern'};
newGraphicScaleFactor = 1000;
sizeSimbolSPC = 40;

tic
%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

% We create the AxesModelHandler

Gs = cell(1);
axesModelHandlerObjects = cell(6,1);

Gs{1} = figure();
set(Gs{1},'Units','inches','Position',[ 10 6 8.5 8 ],'papersize',[8.5 8],'paperposition',[0 0 8.5
8],'color','w','defaultFigurePaperPositionMode','auto');

tic

axesModelHandlerObjects{1} = AxesModelHandler('RW2_TW_Undeformed_S11',Gs{1},[.05 .5
.3 0.5]);
axesModelHandlerObjects{2} = AxesModelHandler('RW2_TW_Undeformed_S22',Gs{1},[.35 .5
.3 0.5]);
axesModelHandlerObjects{3} = AxesModelHandler('RW2_TW_Undeformed_S12',Gs{1},[.65 .5
.3 0.5]);
axesModelHandlerObjects{4} = AxesModelHandler('RW2_TW_Undeformed_E11',Gs{1},[.05 .0
.3 .5]);
axesModelHandlerObjects{5} = AxesModelHandler('RW2_TW_Undeformed_E22',Gs{1},[.35 .0
.3 .5]);
axesModelHandlerObjects{6} = AxesModelHandler('RW2_TW_Undeformed_E12',Gs{1},[.65 .0
.3 .5]);

setBackgroundColor(axesModelHandlerObjects{1},'white')
setBackgroundColor(axesModelHandlerObjects{2},'white')
setBackgroundColor(axesModelHandlerObjects{3},'white')
setBackgroundColor(axesModelHandlerObjects{4},'white')
setBackgroundColor(axesModelHandlerObjects{5},'white')
setBackgroundColor(axesModelHandlerObjects{6},'white')

% We add the AxesModelHandler to the ModelGraphicHandler
```

```

shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {1});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {2});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {3});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {4});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {5});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {6});

% We set the type of ResponseToPlot in each AxesModel
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S11','S11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S22','S22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S12','S12','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E11','E11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E22','E22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E12','E12','Undeformed');

% We set the Scale of the Deformation to draw
shlModelGraphicHandlerObj.setGraphicDeformationScaleFactor(newGraphicScaleFactor);
% We proceed to draw the Last Number of AnalysisStep

for i= Step2Draw
    % First we set the AnalysisStep to Draw
    shlModelGraphicHandlerObj.setAnalysisStepNumber(i);
    % We draw the Elemets

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {1},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {2},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {3},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {4},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {5},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {6},true);
    % We draw the SP_Constraint

```

```

shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {1},sizeSimbolSPC);
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {2},sizeSimbolSPC);
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {3},sizeSimbolSPC);
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {4},sizeSimbolSPC);
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {5},sizeSimbolSPC);
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {6},sizeSimbolSPC);

end
timePerformAnalysis = toc;
fprintf('Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));

```

Ejecutor

```

%We close all the Windows
close all
opt = 0;
if opt==0
    % We clear the memory and the classes and functions
    clear classes;
    clear all;
elseif opt == 1
    % We clear the model, analysis model and modelGraphicHandler
    modelObj.clearAll();
    svlModelResponseRecorderObj.clearAll()
    svlAnalysisModelObj.clearAll()
    shlModelResponseRecorderObj.clearAll()
    shlAnalysisModelObj.clearAll()
    shlModelGraphicHandlerObj.clearAll();
    clear variables
    clear all;
elseif opt == 2
    modelObj.clearAll();
    svlModelResponseRecorderObj.clearAll()
    shlModelResponseRecorderObj.clearAll()
    shlModelGraphicHandlerObj.clearAll();
    clear variables
    clear all;
end
clc

nameFileMat = 'Output_S78_RW2';

% Option to create the Model

% Save or Load Option:
%          1 = Save
%          2 = Load
optionSaveOrLoad = 2;

```

```

% Mesh:
%           1 = Simple Mesh
optionMesh = 1;

% Specimen Wall Option:
%           2 = RW2
optionSpecimenWall = 2;

% Concrete Option:
%           1 =
%           OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01
optionConcrete = 1;

% Steel Option:
%           1 = SmearredEmbeddedSteelDoubleLayerROT2DMaterial01
%           2 = SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01
optionSteel = 2;

% Damage Option:
%           0 = No Damage
%           1 = Damage
optionDamage = 1;

% Type of Element:
%           1 = Quad4N
%           2 = Quad4NR02
optionTypeElement = 2;

% SelfWeight Option:
%           0 = Not Include
%           1 = Include
optionSelfWeight = 1;

%% We create the Model
S78_RW_Model

%% We Create the Static Vertical Load, the Response Recorder and Perform the Analysis
S78_RW_StaticVerticalLoad

%% We Create the Static Horizontal Load, the Response Recorder and Perform the Analysis
S78_RW_StaticHorizontalLoad

%% We create the Graphic for the Static Horizontal Load
iNumberIncrementRecorded_SHL = numberTotalIncrRecorded_SHL-1;

% We extract the Displacement of the Top of the Wall
dispCM =
shlNodeRRObjects {nodeTagToMonitorInTheDisplacementControl}.getResponseRecorded('Displ');

```

```

% We extract the Total Horizontal Load apply at the Top of the Wall
loadCM = numberHorizontalNodeLoads*shlNodeLoadRRObjets {1}.getResponseRecorded();
%% We Create Variable of Displacement
Wall_displ=zeros(numberTotalIncrRecorded_SHL,3);
Wall_displ_Thom=zeros(numberTotalIncrRecorded_SHL,3);
Wall_displ_Tran=zeros(numberTotalIncrRecorded_SHL,3);

Wall_displ(:,1)=dispCM(:,dofNumberHorizontalLoad {1});
Wall_displ_Thom(:,1)=Wall_displ(:,1);
Wall_displ_Tran(:,1)=Wall_displ(:,1);

Wall_load=loadCM(:,dofNumberHorizontalLoad {1})/1000;

%% Graph Step 1%
one=find(Wall_displ(:,1) > height_Wall/100 & Wall_displ(:,1) < height_Wall/100+1);
Step2Draw = one(1);
S78_RW_GraphicInStep

%% Graph Step Max V
Step2Draw=find(Wall_load==max(Wall_load));
S78_RW_GraphicInStep

%% Graphic Element
S78_RW_ElementGraphic

%% Shear and Moment Graphic
S78_ShearFlexion

%%
% We create the Figure that compare the top wall displacement response between the Experiment vs
the Analytical response
fg = figure();
% We plot the graphics
Exp_Results=load('S78_ExperimentalResults.txt');
plot(Exp_Results(:,1),Exp_Results(:,2),'b','LineWidth',0.5);
hold on
plot(Wall_displ(1:2000,1),Wall_load(1:2000),'Color',[1 0 0],'LineWidth',0.5);
xlabel('Desplazamiento [mm]');
ylabel('Fuerza [kN]');
legend('S78','MR5');
grid on
savefig('FiguraAux_S78');

```

Código MR3

Modelo

```

tic
%% Create the Model
modelObj=Model();

nodes2D = load('Nodes_WSH6_RW.txt');
numberNodes = length(nodes2D);
elements2D = load('Elements_WSH6_RW.txt');
numberElements = size(elements2D,1);
nodesSPC = 1 : 17;
horizontalNodeLoads = [307 320 333];
verticalNodeLoads = 347 : 357;
nodeTagToMonitorInTheDisplacementControl = 320;
foundationElements = 1 : 64;
superiorElements= 269 : 316;
boundaryElements_1 = [65 76 77 88 89 100 101
112 113 124 125 136 137 148 149
160 161 172 173 184 185 196 197
208 209 220 221 232 233 244 245
256 257 268];
boundaryElements_2 = [66 75 78 87 90 99 102
111 114 123 126 135 138 147 150
159 162 171 174 183 186 195 198
207 210 219 222 231 234 243 246
255 258 267];
boundaryElements_3 = [67 74 79 86 91 98 103
110 115 122 127 134 139 146 151
158 163 170 175 182 187 194 199
206 211 218 223 230 235 242 247
254 259 266];
centerElements = [68 : 73 80 : 85 92 : 97 104 :
109 116 : 121 128 : 133 140 : 145 152 :
157 164 : 169 176 : 181 188 : 193 200 :
205 212 : 217 224 : 229 236 : 241 248 :
253 260 : 265];

%% Create the Nodes
if optionTypeElement == 1
    numDOF = 2;
else
    numDOF = 3;
end % if

nodesObjects = cell(numberNodes,1);
for i = 1:numberNodes
    nodesObjects {i} = Node(i,numDOF,nodes2D(i,2:3));
end

```

```

    % Add the Nodes that define the element
    modelObj.addNode(nodesObjects{i});
end % for i

%% Material Properties
magnitudGSelfWeightLoad = 9800; % [mm/s^2]
rhoConcreteMaterial = 24.5*(10^-6)/magnitudGSelfWeightLoad; % [N/mm^3] Concrete
nuConcrete = 0.2;

% Steel
%Be Steel - "Confined"
Es_BeVertical = 210300; % [MPa]
Fy_BeVertical = 576; % [MPa]
strainAtFy_BeVertical = Fy_BeVertical/Es_BeVertical; % [mm/mm]

%Vertical Steel
Es_Vertical = 210300; % [MPa]
Fy_Vertical = 583; % [MPa]
strainAtFy_Vertical = Fy_Vertical/Es_Vertical; % [mm/mm]

%Horizontal Steel
Es_Horizontal = 210300; % [MPa]
Fy_Horizontal = 519; % [MPa]
strainAtFy_Horizontal = Fy_Horizontal/Es_Horizontal; % [mm/mm]

%Hoops Steel
Es_Hoops = 210300; % [MPa]
Fy_Hoops = 562; % [MPa]
strainAtFy_Hoops = Fy_Hoops/Es_Hoops; % [mm/mm]

% We add a parameter to the Name of the File to save
nameFileMat = [ nameFileMat '_WR' num2str(optionSpecimenWall) ];

%% Geometry of Wall
% Dimension Rectangular Wall

height_Wall = 4030; % [mm]
thickness_Wall = 150; % [mm]
lengthBoundary_Wall = 385; % [mm]
lengthCenter_Wall = 1230; % [mm]

% We Set the Total Horizontal Force use to Analysis with Displacement
% Control
totalHorizontalForceLoad = 1000; %[N]

```

```

% We define the Total Vertical Load
totalVerticalForceLoad = -1476000; %[N]

%% %%DEFINIDO POR ENSAYOS
% We define the Lateral Drift Procedure
lateralDrift = [0:10 10.5      -7.9 10 -8 26.0      -25.2 26 -25   38.5   -38.7 39 -39   51.8
               -50.2 52 -50   64.7   -64.3 65 -64   77.8   -77.4  78 -77 94   -90.1 -100];%ciclo
%% %%DEFINIDO POR ENSAYOS
%% %%

numberLateralDrift = length(lateralDrift);
time = zeros(numberLateralDrift,1);
numberStepsPerCicle = zeros(numberLateralDrift,1);
totalTime = 0;
slope = 1/1;
for i=1:numberLateralDrift
    lateralDrift(i) = sign(lateralDrift(i))*ceil(abs(lateralDrift(i))*10)/10;
end % for i
for i=1:numberLateralDrift-1
    totalTime = totalTime + ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
    time(i+1) = totalTime;
    numberStepsPerCicle(i+1) = ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
end % for i
fprintf('Number of Steps to Perform is: %s \n',num2str(totalTime));

factorReductionFc = 1.1;

%% %% Confinamiento

% Concrete
Fc_Pedestal = -45; % [MPa]
Ec_Pedestal = 4700*sqrt(-Fc_Pedestal); % [MPa]

Fc = -45.6*factorReductionFc; % [MPa] %46
strainAtFc_Center = -0.003; % [mm/mm]
strainAtFc_Boundary = -0.0055; % [mm/mm]
Fcr = 0.31*sqrt(-Fc); % [MPa]
strainAtFcr = 0.00008; % [mm/mm]

%% %% Cuantias de Armaduras

rho_Hoops_Area_1 = 0.0112;
rho_BeVerticalSteel_Area_1 = 0.0174;
rho_Hoops_Area_2 = 0.0075;
rho_BeVerticalSteel_Area_2 = 0.0054;

```


APÉNDICE B. CÓDIGO DE PROGRAMACIÓN

```
rho_HorizontalSteel_Area= 0.0025;
rho_VerticalSteel_Area = 0.0054;

%% Steel Constant

ratioRhoFyoverFc = abs(rho_Hoops_Area_1*Fy_Hoops/Fc);
Kconf_Opt1 = 1 + ratioRhoFyoverFc; % Orakcal et al. 2006
Kconf_Opt2 = 1 + 0.92*ratioRhoFyoverFc - 0.76*ratioRhoFyoverFc^2; % Vechio 1992 ( Kupfer et
al 1969 )
Kconf_Opt3 = -1.254 + 2.254*sqrt(1+7.94*(ratioRhoFyoverFc)) - 2*(ratioRhoFyoverFc); % Mander
et al (1988)
Kconf = Kconf_Opt1; % Use

%% Create the 2D Material
% Create the Smearred Steel 2D Material
% Steel 2D Material Class Parameters:

if optionSteel == 1
    nameFileMat = [ nameFileMat '_SES' ];
    bsh = 0.01;
    Ac = 1.9;
    Rc = 10;
    OrientationEmbeddedSteel = 0;

    smearedESteel_Area_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerROT2DMaterial01('SmearredESteel_Area',[Es_Horizontal,Es_
    Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[2*rho_HorizontalSteel_Area,2*rho_VerticalSteel_Ar
    ea],OrientationEmbeddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Center,0);
    smearedESteel_AreaBe1_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerROT2DMaterial01('SmearredESteel_AreaBe',[Es_Horizontal,Es_
    _BeVertical],[Fy_Horizontal*rho_HorizontalSteel_Area +
    Fy_Hoops*rho_Hoops_Area_1)/(rho_HorizontalSteel_Area+rho_Hoops_Area_1),Fy_BeVertical],[
    Fcr,Fcr],[2*(rho_HorizontalSteel_Area+rho_Hoops_Area_1),2*rho_BeVerticalSteel_Area_1],Orient
    ationEmbeddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Boundary,0);

elseif optionSteel == 2
    nameFileMat = [ nameFileMat '_SS' ];
    OrientationEmbeddedSteel = 0;
    R0 = 18;
    CR1 = 0.9;
    CR2 = 0.15;
    a1 = -0.002; %To model Damage in the steel
    a2 = 1;
    a3 = -0.002; %To model Damage in the steel
    a4 = 1;
    smearedESteel_Area_2DMatObj =
    SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearredESteel_Area',[Es_Horizontal,Es_
    Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_
```

```

Area,rho_VerticalSteel_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,stra
inAtFc_Center,0);
    smearedESteel_AreaBe1_2DMatObj =
SmearedEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearedESteel_AreaBe',[Es_Horizontal,Es
_BeVertical],[Fy_Horizontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area_2)/(rho_HorizontalSteel_Area+rho_Hoops_Area_2),Fy_BeVertical],[
Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_Area+rho_Hoops_Area_2,rho_BeVert
icalSteel_Area_1],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,strainAtFc_Bo
undary,0);
    smearedESteel_AreaBe2_2DMatObj =
SmearedEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearedESteel_AreaBe',[Es_Horizontal,Es
_BeVertical],[Fy_Horizontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area_1)/(rho_HorizontalSteel_Area+rho_Hoops_Area_1),Fy_BeVertical],[
Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_Area+rho_Hoops_Area_1,rho_BeVert
icalSteel_Area_1],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,strainAtFc_Bo
undary,0);
    smearedESteel_AreaBe3_2DMatObj =
SmearedEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearedESteel_AreaBe',[Es_Horizontal,Es
_BeVertical],[Fy_Horizontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area_2)/(rho_HorizontalSteel_Area+rho_Hoops_Area_2),Fy_BeVertical],[
Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_Area+rho_Hoops_Area_2,rho_BeVert
icalSteel_Area_2],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,strainAtFc_Bo
undary,0);

elseif optionSteel == 3

    nameFileMat = [ nameFileMat '_SSP' ];
    OrientationEmbeddedSteel = 0;
    Bsh=0.01;
    R0 = 18;
    CR1 = 0.9;
    CR2 = 0.15;
    a1 = -0.002; %To model Damage in the steel
    a2 = 1;
    a3 = -0.002; %To model Damage in the steel
    a4 = 1;

    M=38;
    Ecase = 1/300;
    Ehtens = 0.04;
    Eutens = 0.11;
    Sigutens = 630;
    Eohtens = 7000;
    Ef =0.55;
    Ff =0;
    InitialStress = 0;

```

```
steel1UniaxialLayer_Area=SteelUniaxialMaterial('SteelUniaxial_Area',Es_Vertical,Fy_Vertical,Bsh,R0
,CR1,CR2,a1,a2,a3,a4);
```

```
steel2UniaxialLayer_Area=SteelUniaxialMaterial('SteelMMUniaxial_Area',Es_Horizontal,Fy_Horizo
ntal,Bsh,R0,CR1,CR2,a1,a2,a3,a4);
```

```
steel1UniaxialLayer_AreaBe=SteelUniaxialMaterial('SteelUniaxial_AreaBe',Es_BeVertical,(Fy_Horiz
ontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area_1)/(rho_HorizontalSteel_Area+rho_Hoops_Area_1),Bsh,R0,CR1,CR
2,a1,a2,a3,a4);
```

```
steel2UniaxialLayer_AreaBe=SteelMMUniaxialMaterial('SteelMMUniaxial_AreaBe',M,d_BarBeVertic
al,76,Ecase,440,Es_BeVertical,Ehtens,Eutens,Sigutens,Eohtens,Ef,Ff,InitialStress);
```

```
smearESteel_Area_2DMatObj =
SteelDoubleLayer2DMaterial('SmearESteel_Area',rho_HorizontalSteel_Area,steel1UniaxialLayer_Ar
ea,rho_VerticalSteel_Area,steel2UniaxialLayer_Area,OrientationEmbeddedSteel);
```

```
smearESteel_AreaBe1_2DMatObj =
SteelDoubleLayer2DMaterial('SmearESteel_AreaBe',rho_HorizontalSteel_Area+rho_Hoops_Area_
2,steel1UniaxialLayer_AreaBe,rho_BeVerticalSteel_Area_1,steel2UniaxialLayer_AreaBe,Orientation
EmbeddedSteel);
```

```
smearESteel_AreaBe2_2DMatObj =
SteelDoubleLayer2DMaterial('SmearESteel_AreaBe',rho_HorizontalSteel_Area+rho_Hoops_Area_
1,steel1UniaxialLayer_AreaBe,rho_BeVerticalSteel_Area_1,steel2UniaxialLayer_AreaBe,Orientation
EmbeddedSteel);
```

```
smearESteel_AreaBe3_2DMatObj =
SteelDoubleLayer2DMaterial('SmearESteel_AreaBe',rho_HorizontalSteel_Area+rho_Hoops_Area_
2,steel1UniaxialLayer_AreaBe,rho_BeVerticalSteel_Area_2,steel2UniaxialLayer_AreaBe,Orientation
EmbeddedSteel);
```

```
end
```

```
%% Create the Concrete Uniaxial Material
```

```
if optionDamage == 0
```

```
    damageConstantUnconf_1 = 0;
```

```
    damageConstantUnconf_2 = 0;
```

```
    damageConstantConf_1 = 0;
```

```
    damageConstantConf_2 = 0;
```

```
elseif optionDamage == 1
```

```
    damageConstantUnconf_1 = 0.1;
```

```
    damageConstantUnconf_2 = 0.4;
```

```
    damageConstantConf_1 = 0.1;
```

```
    damageConstantConf_2 = 0.4;
```

```
end % if optionConcrete
```

```
%% Create the Orthotropic Rotating Angle 2D Material
```

```

% Tension Stiffening Concrete 2DMaterial
nameFileMat = [ nameFileMat '_TS' ];
K = 1;
AlphaC = 0.32;
AlphaT = 0.08;

% Unconfined
R = 3.1;
B = 0.6;
orthotropicConcrete_Area_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_Area',Fc
,strainAtFc_Center,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstantUnconf_1,da
mageConstantUnconf_2,rhoConcreteMaterial);

% Confined
R = 2.1;
B = 0.4;
orthotropicConcrete_AreaBe_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_AreaCo'
,Kconf*Fc,strainAtFc_Boundary,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstan
tConf_1,damageConstantConf_2,rhoConcreteMaterial);

if optionDamage == 0
elseif optionDamage == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat 'wUD_1_5' ];
end % if optionConcrete

if optionSelfWeight == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_SW' ];
end % if

%% Profile Properties
% We create the profile for the Reinforced Concrete Wall
coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_Area_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_Area',coordinatesConcreteLayers,c
oordinatesReinforcedSteelLayers,thickness_Wall);

coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_AreaConf_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_AreaBe',coordinatesConcreteLayer
s,coordinatesReinforcedSteelLayers,thickness_Wall);

```

```

%Create the Reinforced Concrete Layered Membrane Section
concrete2DMaterialObjectAtEachLayer = {orthotropicConcrete_Area_RA2DMatObj};
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_Area_2DMatObj };
rcWall_Area_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_Area',rcWall_Area_ProfileObj,concr
ete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

concrete2DMaterialObjectAtEachLayer = { orthotropicConcrete_AreaBe_RA2DMatObj };
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_AreaBe1_2DMatObj };
rcWall_AreaConf1_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_AreaConf',rcWall_AreaConf_Profile
Obj,concrete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_AreaBe2_2DMatObj };
rcWall_AreaConf2_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_AreaConf',rcWall_AreaConf_Profile
Obj,concrete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_AreaBe3_2DMatObj };
rcWall_AreaConf3_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_AreaConf',rcWall_AreaConf_Profile
Obj,concrete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

%Create the Elastic Concrete Membrane Pedestal Section

% Create the 2D Material
elasticConcrete2DMat =
ElasticIsotropicPlaneStress2DMaterial('ElasticConcrete2DMat',Ec_Pedestal,nuConcrete,rhoConcret
eMaterial);
thickness_Pedestal = 700; % [mm]
rcWall_Pedestal_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_Pedestal);
rcWall_Pedestal_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_Pedestal_ProfileObj,elasticConcrete2DMat);

%Create the Elastic Concrete Membrane Superior Section
%Create the 2D Material

thickness_SupSection = 400; % [mm]
rcWall_SupSection_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_SupSection);
rcWall_SupSection_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_SupSection_ProfileObj,elasticConcrete2DMat);

%% Create the Element
quad4NObjects = cell(numberElements,1);
if optionTypeElement == 2
    % We add a parameter to the Name of the File to save

```

```

nameFileMat = [ nameFileMat '_Rot' ];
% Elements with Drilling DOF

% We add the Foundation Elements
numberElements_Area = length(foundationElements);
for i = 1:numberElements_Area
    node1 = elements2D(foundationElements(i),2);
    node2 = elements2D(foundationElements(i),3);
    node3 = elements2D(foundationElements(i),4);
    node4 = elements2D(foundationElements(i),5);
    quad4NObjects {foundationElements(i)} = Quad4NR02(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {foundationElements(i)});
end % for i

% We add the Superior Elements
numberElements_Area = length(superiorElements);
for i = 1:numberElements_Area
    node1 = elements2D(superiorElements(i),2);
    node2 = elements2D(superiorElements(i),3);
    node3 = elements2D(superiorElements(i),4);
    node4 = elements2D(superiorElements(i),5);
    quad4NObjects {superiorElements(i)} = Quad4NR02(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {superiorElements(i)});
end % for i

% We add the Boundary Elements 1
numberElements_Area = length(boundaryElements_1);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements_1(i),2);
    node2 = elements2D(boundaryElements_1(i),3);
    node3 = elements2D(boundaryElements_1(i),4);
    node4 = elements2D(boundaryElements_1(i),5);
    quad4NObjects {boundaryElements_1(i)} = Quad4NR02(['E'
num2str(boundaryElements_1(i))],node1,node2,node3,node4,rcWall_AreaConf1_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements_1(i)});
end % for i

% We add the Boundary Elements 2
numberElements_Area = length(boundaryElements_2);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements_2(i),2);

```

```

        node2 = elements2D(boundaryElements_2(i),3);
        node3 = elements2D(boundaryElements_2(i),4);
        node4 = elements2D(boundaryElements_2(i),5);
        quad4NObjects {boundaryElements_2(i)} = Quad4NR02(['E'
num2str(boundaryElements_2(i))],node1,node2,node3,node4,rcWall_AreaConf2_SectionObj,option
TypeElement);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {boundaryElements_2(i)});
    end % for i

    % We add the Boundary Elements 3
    numberElements_Area = length(boundaryElements_3);
    for i = 1:numberElements_Area
        node1 = elements2D(boundaryElements_3(i),2);
        node2 = elements2D(boundaryElements_3(i),3);
        node3 = elements2D(boundaryElements_3(i),4);
        node4 = elements2D(boundaryElements_3(i),5);
        quad4NObjects {boundaryElements_3(i)} = Quad4NR02(['E'
num2str(boundaryElements_3(i))],node1,node2,node3,node4,rcWall_AreaConf3_SectionObj,option
TypeElement);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {boundaryElements_3(i)});
    end % for i

    % We add the Center Elements
    numberElements = length(centerElements);
    for i = 1:numberElements
        node1 = elements2D(centerElements(i),2);
        node2 = elements2D(centerElements(i),3);
        node3 = elements2D(centerElements(i),4);
        node4 = elements2D(centerElements(i),5);
        quad4NObjects {centerElements(i)} = Quad4NR02(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj,optionTypeElemen
t);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {centerElements(i)});
    end % for i

else
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat " ];
    % Elements without Drilling DOF
    % We add the Foundation Elements
    numberElements_Area = length(foundationElements);
    for i = 1:numberElements_Area
        node1 = elements2D(foundationElements(i),2);
        node2 = elements2D(foundationElements(i),3);
        node3 = elements2D(foundationElements(i),4);
        node4 = elements2D(foundationElements(i),5);

```

```

quad4NObjects {foundationElements(i)} = Quad4N(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj);
% Add the Elements that define the element
modelObj.addElement(quad4NObjects {foundationElements(i)});
end % for i

% We add the Foundation Elements
numberElements_Area = length(superiorElements);
for i = 1:numberElements_Area
node1 = elements2D(superiorElements(i),2);
node2 = elements2D(superiorElements(i),3);
node3 = elements2D(superiorElements(i),4);
node4 = elements2D(superiorElements(i),5);
quad4NObjects {superiorElements(i)} = Quad4N(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj);
% Add the Elements that define the element
modelObj.addElement(quad4NObjects {superiorElements(i)});
end % for i

% We add the Boundary Elements 1
numberElements_Area = length(boundaryElements_1);
for i = 1:numberElements_Area
node1 = elements2D(boundaryElements_1(i),2);
node2 = elements2D(boundaryElements_1(i),3);
node3 = elements2D(boundaryElements_1(i),4);
node4 = elements2D(boundaryElements_1(i),5);
quad4NObjects {boundaryElements_1(i)} = Quad4NR02(['E'
num2str(boundaryElements_1(i))],node1,node2,node3,node4,rcWall_AreaConf1_SectionObj,option
TypeElement);
% Add the Elements that define the element
modelObj.addElement(quad4NObjects {boundaryElements_1(i)});
end % for i

% We add the Boundary Elements 2
numberElements_Area = length(boundaryElements_2);
for i = 1:numberElements_Area
node1 = elements2D(boundaryElements_2(i),2);
node2 = elements2D(boundaryElements_2(i),3);
node3 = elements2D(boundaryElements_2(i),4);
node4 = elements2D(boundaryElements_2(i),5);
quad4NObjects {boundaryElements_2(i)} = Quad4NR02(['E'
num2str(boundaryElements_2(i))],node1,node2,node3,node4,rcWall_AreaConf2_SectionObj,option
TypeElement);
% Add the Elements that define the element
modelObj.addElement(quad4NObjects {boundaryElements_2(i)});
end % for i

% We add the Boundary Elements 3
numberElements_Area = length(boundaryElements_3);

```



```

for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements_3(i),2);
    node2 = elements2D(boundaryElements_3(i),3);
    node3 = elements2D(boundaryElements_3(i),4);
    node4 = elements2D(boundaryElements_3(i),5);
    quad4NObjects {boundaryElements_3(i)} = Quad4NR02(['E'
num2str(boundaryElements_3(i))],node1,node2,node3,node4,rcWall_AreaConf3_SectionObj,option
TypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements_3(i)});
end % for i

% We add the Center Elements
numberElements = length(centerElements);
for i = 1:numberElements
    node1 = elements2D(centerElements(i),2);
    node2 = elements2D(centerElements(i),3);
    node3 = elements2D(centerElements(i),4);
    node4 = elements2D(centerElements(i),5);
    quad4NObjects {centerElements(i)} = Quad4N(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {centerElements(i)});
end % for i

end % if optionTypeElement
numberElements = size(elements2D,1);

%% Create the Constraint
numberConstraints = length(nodesSPC);
spcObjects = cell(numberConstraints,1);
for i=1:numberConstraints
    spcObjects {i} = FixRestraint(['SPC_N' num2str(nodesSPC(i))],nodesSPC(i),[1 2 3]);
    % We add the SP_Constraint
    modelObj.addSP_Constraint(spcObjects {i});
end % for i

%% We store the time to take create the Model
timeModel = toc;
fprintf('Time to initialize the Model Object and ModelComponents Objects is %s [sec]
\n',num2str(timeModel));

%% We proceed to set The type of Nodes Reactions to Calculate
typeReactions = 0;
flagNodesReactions = true;
modelObj.setTypeNodesReactions(typeReactions,flagNodesReactions);

```

Generador de Mallado

```
Mesh_fc2(3,2000,4030,11,16,130,125,920,'Nodes_WSH6_RW.txt','Elements_WSH6_RW.txt')
```

Gráfico de Mallado

```
%% Define Variables Depending of the Model
NameAxesModel = {'Modelacion de Muro - MR3'};

%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

% We create the AxesModelHandler

fg{6} = figure();
set(fg{6}, 'Position', [1025 545 1100*1.3 850*1.3], 'papersize', [11 8.5], 'paperposition', [0 0 11 8.5]);
axesModelHandlerObjects{1} = AxesModelHandler(NameAxesModel{1}, fg{6});

% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{1});

% We set the LoadPattern to Draw in the AxesModel 1
shlModelGraphicHandlerObj.setLoadPatternToDraw(NameAxesModel{1}, shlLoadPatternTag, 'Undeformed');
shlModelGraphicHandlerObj.drawElements(axesModelHandlerObjects{1});
shlModelGraphicHandlerObj.drawNodes(axesModelHandlerObjects{1});

shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects{1}, sizeSimbolSPC);
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementShadow')
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementClass')

timePerformAnalysis = toc;
fprintf("Time to perform the Graphic of the Model: %s [sec] \n", num2str(timePerformAnalysis));
```

Gráfico de Tensión y Deformación

```
%% Define Variables Depending of the Model
NameAxesModel
= {'RW2_TW_Undeformed', 'RW2_TW_Deformed', 'RW2_TW_Undeformed_CrackPattern'};
newGraphicScaleFactor = 1000;
sizeSimbolSPC = 40;

tic
%% We create the ModelGraphicHandler
```

```

shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

% We create the AxesModelHandler

Gs = cell(1);
axesModelHandlerObjects = cell(6,1);

Gs{1} = figure();
set(Gs{1},'Units','inches','Position',[ 10 6 12.5 12 ],'papersize',[12.5 12],'paperposition',[0 0 12.5
12],'color','w','defaultFigurePaperPositionMode','auto');

tic

axesModelHandlerObjects{1} = AxesModelHandler('RW2_TW_Undeformed_S11',Gs{1},[.05 .5
.3 0.5]);
axesModelHandlerObjects{2} = AxesModelHandler('RW2_TW_Undeformed_S22',Gs{1},[.35 .5
.3 0.5]);
axesModelHandlerObjects{3} = AxesModelHandler('RW2_TW_Undeformed_S12',Gs{1},[.65 .5
.3 0.5]);
axesModelHandlerObjects{4} = AxesModelHandler('RW2_TW_Undeformed_E11',Gs{1},[.05 .0
.3 .5]);
axesModelHandlerObjects{5} = AxesModelHandler('RW2_TW_Undeformed_E22',Gs{1},[.35 .0
.3 .5]);
axesModelHandlerObjects{6} = AxesModelHandler('RW2_TW_Undeformed_E12',Gs{1},[.65 .0
.3 .5]);

setBackgroundColor(axesModelHandlerObjects{1},'white')
setBackgroundColor(axesModelHandlerObjects{2},'white')
setBackgroundColor(axesModelHandlerObjects{3},'white')
setBackgroundColor(axesModelHandlerObjects{4},'white')
setBackgroundColor(axesModelHandlerObjects{5},'white')
setBackgroundColor(axesModelHandlerObjects{6},'white')

% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{1});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{2});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{3});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{4});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{5});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{6});

% We set the type of ResponseToPlot in each AxesModel
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S11','S11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S22','S22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S12','S12','Undeformed');

```

```

shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E11','E11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E22','E22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E12','E12','Undeformed');

% We set the Scale of the Deformation to draw
shlModelGraphicHandlerObj.setGraphicDeformationScaleFactor(newGraphicScaleFactor);
% We proceed to draw the Last Number of AnalysisStep

for i= Step2Draw
    % First we set the AnalysisStep to Draw
    shlModelGraphicHandlerObj.setAnalysisStepNumber(i);
    % We draw the Elemets

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {1},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {2},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {3},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {4},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {5},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {6},true);
    % We draw the SP_Constraint
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {1},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {2},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {3},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {4},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {5},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {6},sizeSimbolSPC);

end
timePerformAnalysis = toc;
fprintf("Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));

```

Ejecutor

```

%We close all the Windows
close all
opt = 0;
if opt==0
    % We clear the memory and the classes and functions
    clear classes;
    clear all;
elseif opt == 1
    % We clear the model, analysis model and modelGraphicHandler
    modelObj.clearAll();
    svlModelResponseRecorderObj.clearAll()
    svlAnalysisModelObj.clearAll()
    shlModelResponseRecorderObj.clearAll()
    shlAnalysisModelObj.clearAll()
    shlModelGraphicHandlerObj.clearAll();
    clear variables
    clear all;
elseif opt == 2
    modelObj.clearAll();
    svlModelResponseRecorderObj.clearAll()
    shlModelResponseRecorderObj.clearAll()
    shlModelGraphicHandlerObj.clearAll();
    clear variables
    clear all;
end
clc

nameFileMat = 'Output_WSH6_RW2';

% Option to create the Model

% Save or Load Option:
%         1 = Save
%         2 = Load
optionSaveOrLoad = 2;

% Mesh:
%         1 = Simple Mesh
optionMesh = 1;

% Specimen Wall Option:
%         2 = RW2
optionSpecimenWall = 2;

% Concrete Option:
%         1 =
%         OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01
optionConcrete = 1;

```

```

% Steel Option:
%           1 = SmearredEmbeddedSteelDoubleLayerROT2DMaterial01
%           2 = SmearredEmbeddedSteelDoubleLayerMP12DMaterial01
optionSteel = 2;

% Damage Option:
%           0 = No Damage
%           1 = Damage
optionDamage = 1;

% Type of Element:
%           1 = Quad4N
%           2 = Quad4NR02
optionTypeElement = 2;

% SelfWeight Option:
%           0 = Not Include
%           1 = Include
optionSelfWeight = 1;

%% We create the Model
WSH6_RW_Model

%% We Create the Static Vertical Load, the Response Recorder and Perform the Analysis
WSH6_RW_StaticVerticalLoad

%% We Create the Static Horizontal Load, the Response Recorder and Perform the Analysis
WSH6_RW_StaticHorizontalLoad

%% We create the Graphic for the Static Horizontal Load
iNumberIncrementRecorded_SHL = numberTotalIncrRecorded_SHL-1;
%WSH6_RW_StaticHorizontalLoadGraphic

% We extract the Displacement of the Top of the Wall
dispCM =
shlNodeRRObjets {nodeTagToMonitorInTheDisplacementControl}.getResponseRecorded('Displ!');
% We extract the Total Horizontal Load apply at the Top of the Wall
loadCM = numberHorizontalNodeLoads*shlNodeLoadRRObjets {1}.getResponseRecorded();
%% We Create Variable of Displacement
Wall_displ=zeros(numberTotalIncrRecorded_SHL,1);
Wall_displ(:,1)=dispCM(:,dofNumberHorizontalLoad {1});
Wall_load=loadCM(:,dofNumberHorizontalLoad {1})/1000;

%% Graphic Element
WSH6_RW_ElementGraphic

% We create the Figure that compare the top wall displacement response between the Experiment vs
the Analytical response
fg = figure();

```

```
% We plot the graphics
Exp_Results=load('WSH6_ExperimentalResults.txt');
plot(Exp_Results(:,1),Exp_Results(:,2),'b','LineWidth',0.5);
hold on
plot(Wall_displ(:,1),Wall_load,'Color',[1 0 0],'LineWidth',0.5);
xlabel('Desplazamiento [mm]');
ylabel('Fuerza [kN]');
legend('WSH6','MR3');
grid on
savefig('FiguraAux_WSH6');
```

Código MR4

Modelo

```

tic
%% Create the Model
modelObj=Model();
nodes2D = load('Nodes_S6_RW.txt');
numberNodes = length(nodes2D);
elements2D = load('Elements_S6_RW.txt');
numberElements = size(elements2D,1);
beamElements2D = load('ElementsBeam_S6.txt');
numberElementsBeam = size(beamElements2D,1);
nodesSPC = 1 : 19;
horizontalNodeLoads = [351 366 381];
verticalNodeLoads = [397:400 406:409];
nodeTagToMonitorInTheDisplacementControl = 351;
foundationElements = 1 : 72;
superiorElements= 311 : 366;
boundaryElements = [73 : 74 85 : 86 87 : 88
    99 : 100 101 : 102 113 : 114 115 :
    116 127 : 128 129 : 130 141 : 142
    143 : 144 155 : 156 157 : 158 169
    : 170 171 : 172 183 : 184 185 : 186
    197 : 198 199 : 200 211 : 212 213 :
    214 225 : 226 227 : 228 239 : 240
    241 : 242 253 : 254 255 : 256 267
    : 268 269 : 270 281 : 282 283 : 284
    295 : 296 297 : 298 309 : 310];
centerElements = [75 : 84 89 : 98
    103 : 112 117 : 126
    131 : 140 145 :
    154 159 : 168
    173 : 182 187 : 196
    201 : 210 215 : 224
    229 : 238 243 : 252
    257 : 266 271 : 280
    285 : 294 299 : 308];

%% Create the Nodes
if optionTypeElement == 1
    numDOF = 2;
else
    numDOF = 3;
end % if

nodesObjects = cell(numberNodes,1);
for i = 1:numberNodes
    nodesObjects {i} = Node(i,numDOF,nodes2D(i,2:3));
end

```



```

% Add the Nodes that define the element
modelObj.addNode(nodesObjects {i});
end % for i

%% Material Properties
magnitudGSelfWeightLoad = 9800; % [mm/s^2]
rhoConcreteMaterial = 0; % 24.5*(10^-6)/magnitudGSelfWeightLoad; % [N/mm^3] Concrete
nuConcrete = 0.2;

% Steel
% Be Steel - "Confined"
Es_BeVertical = 216000; % [MPa]
Fy_BeVertical = 482; % [MPa]
strainAtFy_BeVertical = Fy_BeVertical/Es_BeVertical; % [mm/mm]
d_BarBeVertical = 15.875; % [mm]
Area_BeVertical = pi*(d_BarBeVertical/2)^2; % [mm^2]

% Vertical Steel
Es_Vertical = 211000; % [MPa]
Fy_Vertical = 507; % [MPa]
strainAtFy_Vertical = Fy_Vertical/Es_Vertical; % [mm/mm]
d_BarVertical = 6.35; % [mm]
Area_Vertical = pi*(d_BarVertical/2)^2; % [mm^2]

% Horizontal Steel
Es_Horizontal = 211000; % [MPa]
Fy_Horizontal = 507; % [MPa]
strainAtFy_Horizontal = Fy_Horizontal/Es_Horizontal; % [mm/mm]
d_BarHorizontal = 6.35; % [mm]
Area_Horizontal = pi*(d_BarHorizontal/2)^2; % [mm^2]

% Hoops Steel
Es_Hoops = 190000; % [MPa]
Fy_Hoops = 440; % [MPa]
strainAtFy_Hoops = Fy_Hoops/Es_Hoops; % [mm/mm]
d_BarHoops = 4.55; % [mm]
Area_Hoops = pi*(d_BarHoops/2)^2; % [mm^2]

% We add a parameter to the Name of the File to save
nameFileMat = [ nameFileMat '_WR' num2str(optionSpecimenWall) ];

%% Geometry of Wall
% Dimension Rectangular Wall

height_Wall = 3048; % [mm]
thickness_Wall = 114; % [mm]
coverConcrete_Wall = 13; % [mm]
lengthBoundary_Wall = 279; % [mm]

```

```

lengthCenter_Wall = 1854; % [mm]

% We Set the Total Horizontal Force use to Analysis with Displacement
% Control
totalHorizontalForceLoad = 10000; %[N]

% We define the Total Vertical Load
totalVerticalForceLoad = -596061; %[N]

%% %%DEFINIDO POR ENSAYOS
% We define the Lateral Drift Procedure
lateralDrift = [0 2.59 -2.39 2.54 -2.49 2.61 -2.49 13.35 -12.62 12.82 -13.05 13.07 -12.85 24.27 -25.16
25.57 -24.52 25.44 -26.68 37.78 -38.64 38.44 -42.63 38.97 -42.7 0 -24.7 0 -24.8 0 -24 0 -24.2 49.33 -
50.42 50.17 -52.17 49.51 -52];
%lateralDrift = [0 5 -5 0 12.7 -12.2 25.7 -24.9 -20:30 38.1 38:-1:-40 -42.1 -40:45 49.5
-50.3 48.4 -43.5];%ciclo
%%DEFINIDO POR ENSAYOS
%%

numberLateralDrift = length(lateralDrift);
time = zeros(numberLateralDrift,1);
numberStepsPerCicle = zeros(numberLateralDrift,1);
totalTime = 0;
slope = 1/1;
for i=1:numberLateralDrift
    lateralDrift(i) = sign(lateralDrift(i))*ceil(abs(lateralDrift(i))*10)/10;
end % for i
for i=1:numberLateralDrift-1
    totalTime = totalTime + ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
    time(i+1) = totalTime;
    numberStepsPerCicle(i+1) = ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
end % for i
fprintf('Number of Steps to Perform is: %s \n',num2str(totalTime));

factorReductionFc = 1;

%% Confinamiento

% Concrete
Fc_Pedestal = -35; % [MPa]
Ec_Pedestal = 4700*sqrt(-Fc_Pedestal); % [MPa]

Fc = -40*factorReductionFc; % [MPa] %%Modificar
strainAtFc_Center = -0.003; % [mm/mm]
strainAtFc_Boundary = -0.0055; % [mm/mm]
Fcr = 0.31*sqrt(-Fc); % [MPa]
strainAtFcr = 0.00008; % [mm/mm]

```

%%% Cuantías de Armaduras

```
rho_Hoops_Area = 0.0084;
rho_BeVerticalSteel_Area = 0.0586;
rho_HorizontalSteel_Area = 0.0054;
rho_VerticalSteel_Area = 0.0056;
```

%%% Steel Constant

```
ratioRhoFyoverFc = abs(rho_Hoops_Area*Fy_Hoops/Fc);
Kconf_Opt1 = 1 + ratioRhoFyoverFc; % Orakcal et al. 2006
Kconf_Opt2 = 1 + 0.92*ratioRhoFyoverFc - 0.76*ratioRhoFyoverFc^2; % Vecchio 1992 (Kupfer et al 1969)
Kconf_Opt3 = -1.254 + 2.254*sqrt(1+7.94*(ratioRhoFyoverFc)) - 2*(ratioRhoFyoverFc); % Mander et al (1988)
Kconf = Kconf_Opt1; % Use
```

%%% Create the 2D Material

% Create the Smeared Steel 2D Material

% Steel 2D Material Class Parameters:

if optionSteel == 1

```
nameFileMat = [ nameFileMat '_SES' ];
bsh = 0.01;
Ac = 1.9;
Rc = 10;
OrientationEmbeddedSteel = 0;
```

```
smearredESteel_Area_2DMatObj =
```

```
SmearedEmbeddedSteelDoubleLayerROT2DMaterial01('SmearedESteel_Area',[Es_Horizontal,Es_Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[2*rho_HorizontalSteel_Area,2*rho_VerticalSteel_Area],OrientationEmbeddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Center,0);
```

```
smearredESteel_AreaBe_2DMatObj =
```

```
SmearedEmbeddedSteelDoubleLayerROT2DMaterial01('SmearedESteel_AreaBe',[Es_Horizontal,Es_BeVertical],[(Fy_Horizontal*rho_HorizontalSteel_Area + Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Fy_BeVertical],[Fcr,Fcr],[2*(rho_HorizontalSteel_Area+rho_Hoops_Area),2*rho_BeVerticalSteel_Area],OrientationEmbeddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Boundary,0);
```

elseif optionSteel == 2

```
nameFileMat = [ nameFileMat '_SS' ];
OrientationEmbeddedSteel = 0;
R0 = 18;
CR1 = 0.9;
CR2 = 0.15;
a1 = -0.002; %To model Damage in the steel
a2 = 1;
```

APÉNDICE B. CÓDIGO DE PROGRAMACIÓN

```
a3 = -0.002; %To model Damage in the steel
a4 = 1;
smearedESteel_Area_2DMatObj =
SmearedEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearedESteel_Area',[Es_Horizontal,Es_
Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_
Area,rho_VerticalSteel_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,str
ainAtFc_Center,0);
smearedESteel_AreaBe_2DMatObj =
SmearedEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearedESteel_AreaBe',[Es_Horizontal,Es
_BeVertical],[(Fy_Horizontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Fy_BeVertical],[Fcr,F
cr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_Area+rho_Hoops_Area,rho_BeVerticalStee
l_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,strainAtFc_Boundary,0);

elseif optionSteel == 3

nameFileMat = [ nameFileMat '_SSP' ];
OrientationEmbeddedSteel = 0;
Bsh=0.01;
R0 = 18;
CR1 = 0.9;
CR2 = 0.15;
a1 = -0.002; %To model Damage in the steel
a2 = 1;
a3 = -0.002; %To model Damage in the steel
a4 = 1;

M=38;
Ecase = 1/300;
Ehtens = 0.04;
Eutens = 0.11;
Sigutens = 630;
Eohtens = 7000;
Ef =0.55;
Ff =0;
InitialStress = 0;

steel1UniaxialLayer_Area=SteelUniaxialMaterial('SteelUniaxial_Area',Es_Vertical,Fy_Vertical,Bsh,R0
,CR1,CR2,a1,a2,a3,a4);

steel2UniaxialLayer_Area=SteelUniaxialMaterial('SteelMMUniaxial_Area',Es_Horizontal,Fy_Horizo
ntal,Bsh,R0,CR1,CR2,a1,a2,a3,a4);

steel1UniaxialLayer_AreaBe=SteelUniaxialMaterial('SteelUniaxial_AreaBe',Es_BeVertical,(Fy_Horiz
ontal*rho_HorizontalSteel_Area +
```

```
Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Bsh,R0,CR1,CR2,a1,a2,a3,a4);
```

```
steel2UniaxialLayer_AreaBe=SteelMMUniaxialMaterial('SteelMMUniaxial_AreaBe',M,d_BarBeVertical,76,Ecase,440,Es_BeVertical,Ehtens,Eutens,Sigutens,Eohtens,Ef,Ff,InitialStress);
```

```
smearESteel_Area_2DMatObj =
SteelDoubleLayer2DMaterial('SmearESteel_Area',rho_HorizontalSteel_Area,steel1UniaxialLayer_Area,rho_VerticalSteel_Area,steel2UniaxialLayer_Area,OrientationEmbeddedSteel);
```

```
smearESteel_AreaBe_2DMatObj =
SteelDoubleLayer2DMaterial('SmearESteel_AreaBe',rho_HorizontalSteel_Area+rho_Hoops_Area,steel1UniaxialLayer_AreaBe,rho_BeVerticalSteel_Area,steel2UniaxialLayer_AreaBe,OrientationEmbeddedSteel);
```

```
end
```

```
%% Create the Concrete Uniaxial Material
```

```
if optionDamage == 0
```

```
    damageConstantUnconf_1 = 0;
```

```
    damageConstantUnconf_2 = 0;
```

```
    damageConstantConf_1 = 0;
```

```
    damageConstantConf_2 = 0;
```

```
elseif optionDamage == 1
```

```
    damageConstantUnconf_1 = 0.1;
```

```
    damageConstantUnconf_2 = 0.1;
```

```
    damageConstantConf_1 = 0.1;
```

```
    damageConstantConf_2 = 0.1;
```

```
end % if optionConcrete
```

```
%% Create Elastic Beam Material
```

```
rho_Concrete=24.5*(10^-6)/magnitudGSelfWeightLoad; % [N/mm^3] Concrete
```

```
ElasticMaterial_Area =
```

```
ElasticUniaxialMaterial('BeamMaterial',Ec_Pedestal,nuConcrete,rho_Concrete);
```

```
%% Create the Orthotropic Rotating Angle 2D Material
```

```
% Tension Stiffening Concrete 2DMaterial
```

```
nameFileMat = [ nameFileMat '_TS' ];
```

```
K = 1;
```

```
AlphaC = 0.32;
```

```
AlphaT = 0.08;
```

```
% Unconfined
```

```
R = 3.1;
```

```
B = 0.6;
```

```
orthotropicConcrete_Area_RA2DMatObj =
```

```
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_Area',Fc,strainAtFc_Center,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstantUnconf_1,damageConstantUnconf_2,rhoConcreteMaterial);
```

```

% Confined
R = 2.1;
B = 0.4;
orthotropicConcrete_AreaBe_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_AreaCo'
,Kconf*Fc,strainAtFc_Boundary,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstan
tConf_1,damageConstantConf_2,rhoConcreteMaterial);

if optionDamage == 0
elseif optionDamage == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat 'wUD_1_5' ];
end % if optionConcrete

if optionSelfWeight == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_SW' ];
end % if

%% Profile Properties
% We create the profile for the Reinforced Concrete Wall
coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_Area_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_Area',coordinatesConcreteLayers,c
oordinatesReinforcedSteelLayers,thickness_Wall);

coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_AreaConf_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_AreaBe',coordinatesConcreteLayer
s,coordinatesReinforcedSteelLayers,thickness_Wall);

%Create the Reinforced Concrete Layered Membrane Section
concrete2DMaterialObjectAtEachLayer = {orthotropicConcrete_Area_RA2DMatObj};
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_Area_2DMatObj };
rcWall_Area_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_Area',rcWall_Area_ProfileObj,concr
ete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

concrete2DMaterialObjectAtEachLayer = { orthotropicConcrete_AreaBe_RA2DMatObj };
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_AreaBe_2DMatObj };
rcWall_AreaConf_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_AreaConf',rcWall_AreaConf_Profile
Obj,concrete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

```

```

%Create the Elastic Concrete Membrane Pedestal Section

% Create the 2D Material
elasticConcrete2DMat =
ElasticIsotropicPlaneStress2DMaterial('ElasticConcrete2DMat',Ec_Pedestal,nuConcrete,rhoConcreteMaterial);
thickness_Pedestal = 660; % [mm]
rcWall_Pedestal_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_Pedestal);
rcWall_Pedestal_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_Pedestal_ProfileObj,elasticConcrete2DMat);

%Create the Elastic Concrete Membrane Superior Section
%Create the 2D Material

thickness_SupSection = 400; % [mm]
rcWall_SupSection_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_SupSection);
rcWall_SupSection_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_SupSection_ProfileObj,elasticConcrete2DMat);

%Create the Elastic Concrete Beam Seccion
%beam
h_b=76; %mm
b_b=2032; %mm
Area= h_b*b_b;
I33=(h_b^3)*(b_b)/12;
Z33=1956*10^3; %mm3
RadOfG33=21.9; %mm
I22=(h_b)*(b_b^3)/12; %mm4
Z22=52300^3; %mm3
RadOfG22=21.9; %mm
J=7433*10^4; %mm4
centroidPos=[0;0]; %mm

BeamProfileObj =
GeneralProfile('BeamProfile',Area,I33,Z33,RadOfG33,I22,Z22,RadOfG22,J,centroidPos);
BeamSectionObj =
ElasticBeamColumnSection('BeamSection',BeamProfileObj,ElasticMaterial_Area);
BeamLineCoord = LinearLineElementCoordTransformation2D('Coords');
%% Create the Element
quad4NObjects = cell(numberElements,1);
beamLineObjects = cell(numberElementsBeam,1);
if optionTypeElement == 2
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_Rot' ];
    % Elements with Drilling DOF

    % We add the Foundation Elements

```

```

numberElements_Area = length(foundationElements);
for i = 1:numberElements_Area
    node1 = elements2D(foundationElements(i),2);
    node2 = elements2D(foundationElements(i),3);
    node3 = elements2D(foundationElements(i),4);
    node4 = elements2D(foundationElements(i),5);
    quad4NObjects {foundationElements(i)} = Quad4NR02(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {foundationElements(i)});
end % for i

% We add the Superior Elements
numberElements_Area = length(superiorElements);
for i = 1:numberElements_Area
    node1 = elements2D(superiorElements(i),2);
    node2 = elements2D(superiorElements(i),3);
    node3 = elements2D(superiorElements(i),4);
    node4 = elements2D(superiorElements(i),5);
    quad4NObjects {superiorElements(i)} = Quad4NR02(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {superiorElements(i)});
end % for i

% We add the Boundary Elements
numberElements_Area = length(boundaryElements);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements(i),2);
    node2 = elements2D(boundaryElements(i),3);
    node3 = elements2D(boundaryElements(i),4);
    node4 = elements2D(boundaryElements(i),5);
    quad4NObjects {boundaryElements(i)} = Quad4NR02(['E'
num2str(boundaryElements(i))],node1,node2,node3,node4,rcWall_AreaConf_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements(i)});
end % for i

% We add the Center Elements
numberElements = length(centerElements);
for i = 1:numberElements
    node1 = elements2D(centerElements(i),2);
    node2 = elements2D(centerElements(i),3);
    node3 = elements2D(centerElements(i),4);
    node4 = elements2D(centerElements(i),5);

```



```

quad4NObjects {centerElements(i)} = Quad4NR02(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj,optionTypeElemen
t);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {centerElements(i)});
end % for i

    % We add the Beam Elements
for i = 1:numberElementsBeam
    node1 = beamElements2D(i,2);
    node2 = beamElements2D(i,3);
    beamLineObjects {i} = ElasticBeamColumn2D(['BE'
num2str(beamElements2D(i))],node1,node2,BeamSectionObj,BeamLineCoord);
    % Add the Elements that define the element
    modelObj.addElement(beamLineObjects {i});
end

else
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat " "];
    % Elements without Drilling DOF
    % We add the Foundation Elements
    numberElements_Area = length(foundationElements);
    for i = 1:numberElements_Area
        node1 = elements2D(foundationElements(i),2);
        node2 = elements2D(foundationElements(i),3);
        node3 = elements2D(foundationElements(i),4);
        node4 = elements2D(foundationElements(i),5);
        quad4NObjects {foundationElements(i)} = Quad4N(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {foundationElements(i)});
    end % for i

    % We add the Foundation Elements
    numberElements_Area = length(superiorElements);
    for i = 1:numberElements_Area
        node1 = elements2D(superiorElements(i),2);
        node2 = elements2D(superiorElements(i),3);
        node3 = elements2D(superiorElements(i),4);
        node4 = elements2D(superiorElements(i),5);
        quad4NObjects {superiorElements(i)} = Quad4N(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {superiorElements(i)});
    end % for i

    % We add the Boundary Elements
    numberElements_Area = length(boundaryElements);

```

```

for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements(i),2);
    node2 = elements2D(boundaryElements(i),3);
    node3 = elements2D(boundaryElements(i),4);
    node4 = elements2D(boundaryElements(i),5);
    quad4NObjects {boundaryElements(i)} = Quad4N(['E'
num2str(boundaryElements(i))],node1,node2,node3,node4,rcWall_AreaConf_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements(i)});
end % for i

% We add the Center Elements
numberElements = length(centerElements);
for i = 1:numberElements
    node1 = elements2D(centerElements(i),2);
    node2 = elements2D(centerElements(i),3);
    node3 = elements2D(centerElements(i),4);
    node4 = elements2D(centerElements(i),5);
    quad4NObjects {centerElements(i)} = Quad4N(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {centerElements(i)});
end % for i

% We add the Beam Elements
for i = 1:numberElementsBeam
    node1 = beamElements2D(i,2);
    node2 = beamElements2D(i,3);
    beamLineObjects {i} = ElasticBeamColumn2D(['BE'
num2str(beamElements2D(i))],node1,node2,BeamSectionObj,BeamLineCoord);
    % Add the Elements that define the element
    modelObj.addElement(beamLineObjects {i});
end

end % if optionTypeElement
numberElements = size(elements2D,1);

%% Create the Constraint
numberConstraints = length(nodesSPC);
spcObjects = cell(numberConstraints,1);
for i=1:numberConstraints
    spcObjects {i} = FixRestraint(['SPC_N' num2str(nodesSPC(i))],nodesSPC(i),[1 2 3]);
    % We add the SP_Constraint
    modelObj.addSP_Constraint(spcObjects {i});
end % for i

%% We store the time to take create the Model

```

```
timeModel = toc;
fprintf("Time to initialize the Model Object and ModelComponents Objects is %s [sec]
\n',num2str(timeModel));
```

```
%% We proceed to set The type of Nodes Reactions to Calculate
typeReactions = 0;
flagNodesReactions = true;
modelObj.setTypeNodesReactions(typeReactions,flagNodesReactions);
```

Generador de Mallado

```
Mesh_fc2(2,2412,3123,13,16,130,137,600,'Nodes_S6_RW.txt','Elements_S6_RW.txt')
```

Gráfico de Mallado

```
%% Define Variables Depending of the Model
NameAxesModel = {'Modelaci?n de Muro - MR4'};

%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

% We create the AxesModelHandler

MESH=figure();
set(MESH,'Position',[ 1025 545 1100*1.3 850*1.3 ],'papersize',[11 8.5],'paperposition',[0 0 11 8.5]);
axesModelHandlerObjects{1} = AxesModelHandler(NameAxesModel{1},MESH);%,[.05 .05 .4
.4]);

% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{1});

% We set the LoadPattern to Draw in the AxesModel 1
shlModelGraphicHandlerObj.setLoadPatternToDraw(NameAxesModel{1},shlLoadPatternTag,'Undeformed');
shlModelGraphicHandlerObj.drawElements(axesModelHandlerObjects{1});
shlModelGraphicHandlerObj.drawNodes(axesModelHandlerObjects{1});
%shlModelGraphicHandlerObj.drawNodeTags(axesModelHandlerObjects{1});
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects{1},sizeSimbolSPC);
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementShadow')
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementClass')

timePerformAnalysis = toc;
fprintf("Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));
%title('Modelo Muro MR4');
```

Gráfico de Tensión y Deformación

```

%% Define Variables Depending of the Model
NameAxesModel
= {'RW2_TW_Undeformed','RW2_TW_Deformed','RW2_TW_Undeformed_CrackPattern'};
newGraphicScaleFactor = 1000;
sizeSimbolSPC = 40;

tic
%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

% We create the AxesModelHandler

Gs = cell(1);
axesModelHandlerObjects = cell(6,1);

Gs{1} = figure();
set(Gs{1},'Units','inches','Position',[ 10 6 12.5 12 ],'papersize',[12.5 12],'paperposition',[0 0 12.5
12],'color','w','defaultFigurePaperPositionMode','auto');

tic

axesModelHandlerObjects{1} = AxesModelHandler('RW2_TW_Undeformed_S11',Gs{1},[.05 .5
.3 0.5]);
axesModelHandlerObjects{2} = AxesModelHandler('RW2_TW_Undeformed_S22',Gs{1},[.35 .5
.3 0.5]);
axesModelHandlerObjects{3} = AxesModelHandler('RW2_TW_Undeformed_S12',Gs{1},[.65 .5
.3 0.5]);
axesModelHandlerObjects{4} = AxesModelHandler('RW2_TW_Undeformed_E11',Gs{1},[.05 .0
.3 .5]);
axesModelHandlerObjects{5} = AxesModelHandler('RW2_TW_Undeformed_E22',Gs{1},[.35 .0
.3 .5]);
axesModelHandlerObjects{6} = AxesModelHandler('RW2_TW_Undeformed_E12',Gs{1},[.65 .0
.3 .5]);

setBackgroundcolor(axesModelHandlerObjects{1},'white')
setBackgroundcolor(axesModelHandlerObjects{2},'white')
setBackgroundcolor(axesModelHandlerObjects{3},'white')
setBackgroundcolor(axesModelHandlerObjects{4},'white')
setBackgroundcolor(axesModelHandlerObjects{5},'white')
setBackgroundcolor(axesModelHandlerObjects{6},'white')

% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{1});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{2});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{3});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{4});

```

```

shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{5});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects{6});

% We set the type of ResponseToPlot in each AxesModel
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S11','S11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S22','S22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S12','S12','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E11','E11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E22','E22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E12','E12','Undeformed');

% We set the Scale of the Deformation to draw
shlModelGraphicHandlerObj.setGraphicDeformationScaleFactor(newGraphicScaleFactor);
% We proceed to draw the Last Number of AnalysisStep

for i= Step2Draw
    % First we set the AnalysisStep to Draw
    shlModelGraphicHandlerObj.setAnalysisStepNumber(i);
    % We draw the Elements

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects{1},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects{2},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects{3},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects{4},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects{5},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects{6},true);
    % We draw the SP_Constraint
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects{1},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects{2},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects{3},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects{4},sizeSimbolSPC);

```

```
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {5},sizeSymbolSPC);
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {6},sizeSymbolSPC);
```

```
end
timePerformAnalysis = toc;
fprintf("Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));
```

Ejecutor

```
%We close all the Windows
close all
opt = 0;
if opt==0
    % We clear the memory and the classes and functions
    clear classes;
    clear all;
elseif opt == 1
    % We clear the model, analysis model and modelGraphicHandler
    modelObj.clearAll();
    svlModelResponseRecorderObj.clearAll()
    svlAnalysisModelObj.clearAll()
    shlModelResponseRecorderObj.clearAll()
    shlAnalysisModelObj.clearAll()
    shlModelGraphicHandlerObj.clearAll();
    clear variables
    clear all;
elseif opt == 2
    modelObj.clearAll();
    svlModelResponseRecorderObj.clearAll()
    shlModelResponseRecorderObj.clearAll()
    shlModelGraphicHandlerObj.clearAll();
    clear variables
    clear all;
end
clc

nameFileMat = 'Output_S6_RW2';

% Option to create the Model

% Save or Load Option:
%         1 = Save
%         2 = Load
optionSaveOrLoad = 2;

% Mesh:
%         1 = Simple Mesh
optionMesh = 1;
```

```

% Specimen Wall Option:
%           2 = RW2
optionSpecimenWall = 2;

% Concrete Option:
%           1 =
%           OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01
optionConcrete = 1;

% Steel Option:
%           1 = SmearEmbeddedSteelDoubleLayerROT2DMaterial01
%           2 = SmearEmbeddedSteelDoubleLayerMPT2DMaterial01
optionSteel = 2;

% Damage Option:
%           0 = No Damage
%           1 = Damage
optionDamage = 1;

% Type of Element:
%           1 = Quad4N
%           2 = Quad4NR02
optionTypeElement = 2;

% SelfWeight Option:
%           0 = Not Include
%           1 = Include
optionSelfWeight = 0;

%% We create the Model
S6_RW_Model

%% We Create the Static Vertical Load, the Response Recorder and Perform the Analysis
S6_RW_StaticVerticalLoad

%% We Create the Static Horizontal Load, the Response Recorder and Perform the Analysis
S6_RW_StaticHorizontalLoad

%% Remove the beam elements
% We remove the Beam Elements for graphic
for i = 1:numberElementsBeam
    modelObj.removeElement(['BE' num2str(beamElements2D(i))]);
end

%% We create the Graphic for the Static Horizontal Load
iNumberIncrementRecorded_SHL = numberTotalIncrRecorded_SHL-1;

% We extract the Displacement of the Top of the Wall

```

```

dispCM =
shlNodeRRObjets {nodeTagToMonitorInTheDisplacementControl}.getResponseRecorded('Displ');
% We extract the Total Horizontal Load apply at the Top of the Wall
loadCM = numberHorizontalNodeLoads*shlNodeLoadRRObjets {1}.getResponseRecorded();
%% We Create Variable of Displacement
Wall_displ=zeros(numberTotalIncrRecorded_SHL,1);
Wall_displ(:,1)=dispCM(:,dofNumberHorizontalLoad {1});
Wall_load=loadCM(:,dofNumberHorizontalLoad {1})/1000;

%% %% Graph Step 1%
one=find(Wall_displ(:,1) > height_Wall/100 & Wall_displ(:,1) < height_Wall/100+1);
Step2Draw = one(1);
S6_RW_GraphicInStep

%% Graph Step Max V
Step2Draw=find(Wall_load==max(Wall_load));
S6_RW_GraphicInStep

%% Graphic Element
S6_RW_ElementGraphic

%% We create the Figure that compare the top wall displacement response between the Experiment
vs the Analytical response
fg = figure();
set(gca,'box','on')
% We plot the graphics
Exp_Results=load('S6_ExperimentalResults.txt');
plot(Exp_Results(:,1),Exp_Results(:,2),'b','LineWidth',0.5);
hold on
plot(Wall_displ(:,1),Wall_load,'Color',[1 0 0],'LineWidth',0.5);
xlabel('flexi/?n Desplazamiento [mm]');
ylabel('Fuerza [kN]');
legend('S6','MR4');
grid on
savefig('FiguraAux_S6');

```


Código MR5

Modelo

```

tic
%% Create the Model
modelObj=Model();

nodes2D = load('Nodes_R2_RW.txt');
numberNodes = length(nodes2D);
elements2D = load('Elements_R2_RW.txt');
numberElements = size(elements2D,1);
nodesSPC = 1 : 18;
horizontalNodeLoads = [471 487 503];
verticalNodeLoads = 520 : 533;
nodeTagToMonitorInTheDisplacementControl = 471;
nodes_shearDisp=[63 73 120 130]; %Nodos de medicion del corte
foundationElements = 1 : 68;
superiorElements= 420:479;
boundaryElements = [69 70 80 81 82 83 93 94
                    95 96 106 107 108 109 119 120 121 122
                    132 133 134 135 145 146 147 148 158 159
                    160 161 171 172 173 174 184 185 186 187
                    197 198 199 200 210 211];
boundaryElements2 = [212 213 223 224 225 226 236 237
                    238 239 249 250 251 252 262 263 264 265
                    275 276 277 278 288 289 290 291 301 302
                    303 304 314 315 316 317 327 328 329 330
                    340 341 342 343 353 354 355 356 366 367
                    368 369 379 380 381 382 392 393 394 395
                    405 406 407 408 418 419];
centerElements = [71 : 79 84 : 92 97
                  : 105 110 : 118 123 : 131
                  : 136 : 144 149 : 157 162
                  : 170 175 : 183 188 : 196
                  : 201 : 209 214 : 222 227
                  : 235 : 240 248 : 253 261
                  : 266 : 274 279 : 287 292
                  : 300 305 : 313 318 : 326
                  : 331 : 339 344 : 352 357
                  : 365 370 : 378 383 : 391
                  396 : 404 409 : 417];

%% Create the Nodes
if optionTypeElement == 1
    numDOF = 2;
else
    numDOF = 3;
end % if

```

```

nodesObjects = cell(numberNodes,1);
for i = 1:numberNodes
    nodesObjects {i} = Node(i,numDOF,nodes2D(i,2:3));
    % Add the Nodes that define the element
    modelObj.addNode(nodesObjects {i});
end % for i

%% Material Properties
magnitudGSelfWeightLoad = 9800; % [mm/s^2]
rhoConcreteMaterial = 24.5*(10^-6)/magnitudGSelfWeightLoad; % [N/mm^3] Concrete
nuConcrete = 0.2;

% Steel
%Be Steel - "Confined"
Es_BeVertical = 185469; % [MPa]
Fy_BeVertical = 455; % [MPa]
strainAtFy_BeVertical = Fy_BeVertical/Es_BeVertical; % [mm/mm]
d_BarBeVertical = 12.7; % [mm]

%Vertical Steel
Es_Vertical = 224777; % [MPa]
Fy_Vertical = 455; % [MPa]
strainAtFy_Vertical = Fy_Vertical/Es_Vertical; % [mm/mm]
d_BarVertical = 6.3; % [mm]

%Horizontal Steel
Es_Horizontal = 224777; % [MPa]
Fy_Horizontal = 455; % [MPa]
strainAtFy_Horizontal = Fy_Horizontal/Es_Horizontal; % [mm/mm]
d_BarHorizontal = 6.3; % [mm]

%Hoops Steel
Es_Hoops = 195123; % [MPa]
Fy_Hoops = 485; % [MPa]
strainAtFy_Hoops = Fy_Hoops/Es_Hoops; % [mm/mm]
d_BarHoops = 4.95; % [mm]

% We add a parameter to the Name of the File to save
nameFileMat = [ nameFileMat '_WR' num2str(optionSpecimenWall) ];

%% Geometry of Wall
% Dimension Rectangular Wall

height_Wall = 4560; % [mm]
thickness_Wall = 102; % [mm]
coverConcrete_Wall = 25.4-d_BarHoops-d_BarBeVertical/2; % [mm]
lengthBoundary_Wall = 190.5; % [mm]
lengthCenter_Wall = 1524; % [mm]

```

```

% We Set the Total Horizontal Force use to Analysis with Displacement
% Control
totalHorizontalForceLoad = 10000; %[N]

% We define the Total Vertical Load
totalVerticalForceLoad = -2300*9.8*(1219*200*2200+102*4560*1900)/(1000*1000*1000); %[N]

%% %%DEFINIDO POR ENSAYOS
% We define the Lateral Drift Procedure
lateralDrift = [0 1 -1 1 -1 1 -1 2.5 -2.5 2.5 -2.5 2.5 -2.5 1 -1 1 -1 1 -1 5 -5 5 -5 5 -5 12.5 -12.5 12.5 -
12.5 12.5 -12.5 7 -7 7 -7 7 -7 26.2 -26.2 25.4 -25.4 26.2 -24.9 51.3 -49.1 51.1 -49.3
51.0 -49.2 77.3 -75.1 77.1 -75.1 76.9 -74.5 26.2 -24.9 26.2 -24.9
26.2 -24.9 102.1 -101.3 102.0 -100.3 103.5 -101.5 127.5 -128.3 127.6 -125.6 127.6
-126.1 0 153.1 0 -152.1 0 155.0 0 -115.9 -153.5 0 156.5 0 -154.4 0 175 0 -
175];
%% %%DEFINIDO POR ENSAYOS
%%

numberLateralDrift = length(lateralDrift);
time = zeros(numberLateralDrift,1);
numberStepsPerCicle = zeros(numberLateralDrift,1);
totalTime = 0;
slope = 1/1;
for i=1:numberLateralDrift
    lateralDrift(i) = sign(lateralDrift(i))*ceil(abs(lateralDrift(i))*10)/10;
end % for i
for i=1:numberLateralDrift-1
    totalTime = totalTime + ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
    time(i+1) = totalTime;
    numberStepsPerCicle(i+1) = ceil(slope*abs(lateralDrift(i+1)-lateralDrift(i)));
end % for i
fprintf('Number of Steps to Perform is: %s \n',num2str(totalTime));

factorReductionFc = 1;

%% %% Confinamiento

% Concrete
Fc_Pedestal = -45; % [MPa]
Ec_Pedestal = 4700*sqrt(-Fc_Pedestal); % [MPa]

Fc = -50*factorReductionFc; % [MPa] %% Modificar
strainAtFc_Center = -0.003; % [mm/mm]
strainAtFc_Boundary = -0.0055; % [mm/mm]
Fcr = 0.31*sqrt(-Fc); % [MPa]
strainAtFcr = 0.00008; % [mm/mm]

```

```

%% Cuantias

rho_Hoops_Area = 0.0184;
rho_HorizontalSteel_Area = 0.0031;
rho_VerticalSteel_Area = 0.0025;
rho_BeVerticalSteel_Area = 0.0393;

%% Steel + Cuantia Hoops

ratioRhoFyoverFc = abs(rho_Hoops_Area*Fy_Hoops/Fc);
Kconf_Opt1 = 1 + ratioRhoFyoverFc; % Orakcal et al. 2006
Kconf_Opt2 = 1 + 0.92*ratioRhoFyoverFc - 0.76*ratioRhoFyoverFc^2; % Vechio 1992 (Kupfer et al 1969)
Kconf_Opt3 = -1.254 + 2.254*sqrt(1+7.94*(ratioRhoFyoverFc)) - 2*(ratioRhoFyoverFc); % Mander et al (1988)
ke = 0.65;
Kconf = Kconf_Opt1; % Use

%% Create the 2D Material
% Create the Smeared Steel 2D Material
% Steel 2D Material Class Parameters:

if optionSteel == 1
    nameFileMat = [ nameFileMat '_SES' ];
    bsh = 0.01;
    Ac = 1.9;
    Rc = 10;
    OrientationEmbeddedSteel = 0;

    smearedESteel_Area_2DMatObj =
    SmearedEmbeddedSteelDoubleLayerROT2DMaterial01('SmearedESteel_Area',[Es_Horizontal,Es_V
    Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[rho_HorizontalSteel_Area,rho_VerticalSteel_Area],
    OrientationEmbeddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Center,0);
    smearedESteel_AreaBe_2DMatObj =
    SmearedEmbeddedSteelDoubleLayerROT2DMaterial01('SmearedESteel_AreaBe',[Es_Horizontal,Es
    _BeVertical],[(Fy_Horizontal*rho_HorizontalSteel_Area +
    Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Fy_BeVertical],[Fcr,F
    cr],[(rho_HorizontalSteel_Area+rho_Hoops_Area),rho_BeVerticalSteel_Area],OrientationEmbedde
    dSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Boundary,0);
    smearedESteel_AreaBe2_2DMatObj =
    SmearedEmbeddedSteelDoubleLayerROT2DMaterial01('SmearedESteel_AreaBe2',[Es_Horizontal,
    Es_BeVertical],[Fy_Horizontal,Fy_BeVertical],[Fcr,Fcr],[rho_HorizontalSteel_Area,rho_BeVerticalS
    teel_Area],OrientationEmbeddedSteel,[bsh,bsh],[Ac,Ac],[Rc,Rc],strainAtFcr,strainAtFc_Boundary,0);

elseif optionSteel == 2
    nameFileMat = [ nameFileMat '_SS' ];
    OrientationEmbeddedSteel = 0;
    R0 = 18;

```

```

CR1 = 0.9;
CR2 = 0.15;
a1 = -0.002; %To model Damage in the steel
a2 = 1;
a3 = -0.002; %To model Damage in the steel
a4 = 1;
smearredESteel_Area_2DMatObj =
SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearredESteel_Area',[Es_Horizontal,Es_
Vertical],[Fy_Horizontal,Fy_Vertical],[Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_
Area,rho_VerticalSteel_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,str
ainAtFc_Center,0);
smearredESteel_AreaBe_2DMatObj =
SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearredESteel_AreaBe',[Es_Horizontal,Es
_BeVertical],[Fy_Horizontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Fy_BeVertical],[Fcr,F
cr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_HorizontalSteel_Area+rho_Hoops_Area],rho_BeVerticalSte
el_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],strainAtFcr,strainAtFc_Boundary,0
);
smearredESteel_AreaBe2_2DMatObj =
SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01('SmearredESteel_AreaBe2',[Es_Horizontal,
Es_BeVertical],[Fy_Horizontal,Fy_BeVertical],[Fcr,Fcr],[R0,R0],[CR1,CR1],[CR2,CR2],[rho_Horizo
ntalSteel_Area,rho_BeVerticalSteel_Area],OrientationEmbeddedSteel,[a1,a1],[a2,a2],[a3,a3],[a4,a4],str
ainAtFcr,strainAtFc_Boundary,0);

elseif optionSteel == 3
nameFileMat = [ nameFileMat '_SSP' ];
OrientationEmbeddedSteel = 0;
Bsh=0.01;
R0 = 18;
CR1 = 0.9;
CR2 = 0.15;
a1 = -0.002; %To model Damage in the steel
a2 = 1;
a3 = -0.002; %To model Damage in the steel
a4 = 1;

M=38;
Ecase = 1/300;
Ehtens = 0.04;
Eutens = 0.11;
Sigutens = 630;
Eohtens = 7000;
Ef =0.55;
Ff =0;
InitialStress = 0;

steel1UniaxialLayer_Area=SteelUniaxialMaterial('SteelUniaxial_Area',Es_Vertical,Fy_Vertical,Bsh,R0
,CR1,CR2,a1,a2,a3,a4);

```

```

steel2UniaxialLayer_Area=SteelUniaxialMaterial('SteelMMUniaxial_Area',Es_Horizontal,Fy_Horizo
ntal,Bsh,R0,CR1,CR2,a1,a2,a3,a4);

steel1UniaxialLayer_AreaBe=SteelUniaxialMaterial('SteelUniaxial_AreaBe',Es_BeVertical,(Fy_Horiz
ontal*rho_HorizontalSteel_Area +
Fy_Hoops*rho_Hoops_Area)/(rho_HorizontalSteel_Area+rho_Hoops_Area),Bsh,R0,CR1,CR2,a1,a
2,a3,a4);

steel2UniaxialLayer_AreaBe=SteelMMUniaxialMaterial('SteelMMUniaxial_AreaBe',M,d_BarBeVertic
al,76,Ecase,440,Es_BeVertical,Ehtens,Eutens,Sigutens,Eohtens,Ef,Ff,InitialStress);

    smearedESteel_Area_2DMatObj =
SteelDoubleLayer2DMaterial('SmearedSteel_Area',2*rho_HorizontalSteel_Area,steel1UniaxialLayer_
Area,2*rho_VerticalSteel_Area,steel2UniaxialLayer_Area,OrientationEmbeddedSteel);
    smearedESteel_AreaBe_2DMatObj =
SteelDoubleLayer2DMaterial('SmearedSteel_AreaBe',2*(rho_HorizontalSteel_Area+rho_Hoops_Ar
ea),steel1UniaxialLayer_AreaBe,rho_BeVerticalSteel_Area,steel2UniaxialLayer_AreaBe,OrientationE
mbeddedSteel);
    smearedESteel_AreaBe2_2DMatObj =
SteelDoubleLayer2DMaterial('SmearedSteel_AreaBe2',rho_HorizontalSteel_Area,steel1UniaxialLayer
_AreaBe,rho_BeVerticalSteel_Area,steel2UniaxialLayer_AreaBe,OrientationEmbeddedSteel);

end

%% Create the Concrete Uniaxial Material
if optionDamage == 0
    damageConstantUnconf_1 = 0;
    damageConstantUnconf_2 = 0;
    damageConstantConf_1 = 0;
    damageConstantConf_2 = 0;
elseif optionDamage == 1
    damageConstantUnconf_1 = 0.125;
    damageConstantUnconf_2 = 0.5;
    damageConstantConf_1 = 0.125;
    damageConstantConf_2 = 0.5;
end % if optionConcrete

%% Create the Orthotropic Rotating Angle 2D Material

% Tension Stiffening Concrete 2DMaterial
nameFileMat = [ nameFileMat '_TS' ];
K = 1;
AlphaC = 0.32;
AlphaT = 0.08;

% Unconfined
R = 3.1;

```

```

B = 0.6;
orthotropicConcrete_Area_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_Area',Fc
,strainAtFc_Center,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstantUnconf_1,da
mageConstantUnconf_2,rhoConcreteMaterial);

% Confined
R = 2.1;
B = 0.4;
orthotropicConcrete_AreaBe_RA2DMatObj =
OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01('OrthotropicConcrete_AreaCo'
,Kconf*Fc,strainAtFc_Boundary,R,K,AlphaC,Fcr,strainAtFcr,B,AlphaT,nuConcrete,damageConstan
tConf_1,damageConstantConf_2,rhoConcreteMaterial);

if optionDamage == 0
elseif optionDamage == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat 'wUD_1_5' ];
end % if optionConcrete

if optionSelfWeight == 1
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_SW' ];
end % if

%% Profile Properties
% We create the profile for the Reinforced Concrete Wall
coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_Area_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_Area',coordinatesConcreteLayers,c
oordinatesReinforcedSteelLayers,thickness_Wall);

coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_AreaConf_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_AreaBe',coordinatesConcreteLayer
s,coordinatesReinforcedSteelLayers,thickness_Wall);

coordinatesConcreteLayers = [ 0 thickness_Wall ];
coordinatesReinforcedSteelLayers = (thickness_Wall/2);
rcWall_AreaConf2_ProfileObj =
ReinforcedConcreteLayeredPlaneElementProfile('RCWallProfile_AreaBe',coordinatesConcreteLayer
s,coordinatesReinforcedSteelLayers,thickness_Wall);

%Create the Reinforced Concrete Layered Membrane Section
concrete2DMaterialObjectAtEachLayer = {orthotropicConcrete_Area_RA2DMatObj};

```

```

reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_Area_2DMatObj };
rcWall_Area_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_Area',rcWall_Area_ProfileObj,concr
ete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

concrete2DMaterialObjectAtEachLayer = { orthotropicConcrete_AreaBe_RA2DMatObj };
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_AreaBe_2DMatObj };
rcWall_AreaConf_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_AreaConf',rcWall_AreaConf_Profile
Obj,concrete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

concrete2DMaterialObjectAtEachLayer = { orthotropicConcrete_Area_RA2DMatObj };
reinforcedSteel2dMaterialObjectAtEachLayer = { smearedESteel_AreaBe2_2DMatObj };
rcWall_AreaConf2_SectionObj =
ReinforcedConcreteLayerMembraneSection01('RCWallSection_AreaConf2',rcWall_AreaConf2_Prof
ileObj,concrete2DMaterialObjectAtEachLayer,reinforcedSteel2dMaterialObjectAtEachLayer);

%Create the Elastic Concrete Membrane Pedestal Section

% Create the 2D Material
elasticConcrete2DMat =
ElasticIsotropicPlaneStress2DMaterial('ElasticConcrete2DMat',Ec_Pedestal,nuConcrete,rhoConcret
eMaterial);
thickness_Pedestal = 1219; % [mm]
rcWall_Pedestal_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_Pedestal);
rcWall_Pedestal_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_Pedestal_ProfileObj,elasticConcrete2DMat);

%Create the Elastic Concrete Membrane Superior Section
%Create the 2D Material

thickness_SupSection = thickness_Wall;%1219; % [mm]
rcWall_SupSection_ProfileObj =
RectangularPlaneElementProfile('RCWallProfile_Pedestal',thickness_SupSection);
rcWall_SupSection_SectionObj =
PlaneStressSection('RCWallSection_Pedestal',rcWall_SupSection_ProfileObj,elasticConcrete2DMat);

%% Create the Element
quad4NObjects = cell(numberElements,1);
if optionTypeElement == 2
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat '_Rot' ];
    % Elements with Drilling DOF

    % We add the Foundation Elements
    numberElements_Area = length(foundationElements);
    for i = 1:numberElements_Area

```



```

node1 = elements2D(foundationElements(i),2);
node2 = elements2D(foundationElements(i),3);
node3 = elements2D(foundationElements(i),4);
node4 = elements2D(foundationElements(i),5);
quad4NObjects { foundationElements(i) } = Quad4NR02(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects { foundationElements(i) });
end % for i

% We add the Superior Elements
numberElements_Area = length(superiorElements);
for i = 1:numberElements_Area
    node1 = elements2D(superiorElements(i),2);
    node2 = elements2D(superiorElements(i),3);
    node3 = elements2D(superiorElements(i),4);
    node4 = elements2D(superiorElements(i),5);
    quad4NObjects { superiorElements(i) } = Quad4NR02(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects { superiorElements(i) });
end % for i

% We add the Boundary Elements
numberElements_Area = length(boundaryElements);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements(i),2);
    node2 = elements2D(boundaryElements(i),3);
    node3 = elements2D(boundaryElements(i),4);
    node4 = elements2D(boundaryElements(i),5);
    quad4NObjects { boundaryElements(i) } = Quad4NR02(['E'
num2str(boundaryElements(i))],node1,node2,node3,node4,rcWall_AreaConf_SectionObj,optionTypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects { boundaryElements(i) });
end % for i

% We add the Boundary Elements 2
numberElements_Area = length(boundaryElements2);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements2(i),2);
    node2 = elements2D(boundaryElements2(i),3);
    node3 = elements2D(boundaryElements2(i),4);
    node4 = elements2D(boundaryElements2(i),5);
    quad4NObjects { boundaryElements2(i) } = Quad4NR02(['E'
num2str(boundaryElements2(i))],node1,node2,node3,node4,rcWall_AreaConf2_SectionObj,optionTypeElement);

```

```

    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements2(i)});
end % for i

% We add the Center Elements
numberElements = length(centerElements);
for i = 1:numberElements
    node1 = elements2D(centerElements(i),2);
    node2 = elements2D(centerElements(i),3);
    node3 = elements2D(centerElements(i),4);
    node4 = elements2D(centerElements(i),5);
    quad4NObjects {centerElements(i)} = Quad4NR02(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj,optionTypeElemen
t);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {centerElements(i)});
end % for i

else
    % We add a parameter to the Name of the File to save
    nameFileMat = [ nameFileMat " "];
    % Elements without Drilling DOF
    % We add the Foundation Elements
    numberElements_Area = length(foundationElements);
    for i = 1:numberElements_Area
        node1 = elements2D(foundationElements(i),2);
        node2 = elements2D(foundationElements(i),3);
        node3 = elements2D(foundationElements(i),4);
        node4 = elements2D(foundationElements(i),5);
        quad4NObjects {foundationElements(i)} = Quad4N(['E'
num2str(foundationElements(i))],node1,node2,node3,node4,rcWall_Pedestal_SectionObj);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {foundationElements(i)});
    end % for i

    % We add the Foundation Elements
    numberElements_Area = length(superiorElements);
    for i = 1:numberElements_Area
        node1 = elements2D(superiorElements(i),2);
        node2 = elements2D(superiorElements(i),3);
        node3 = elements2D(superiorElements(i),4);
        node4 = elements2D(superiorElements(i),5);
        quad4NObjects {superiorElements(i)} = Quad4N(['E'
num2str(superiorElements(i))],node1,node2,node3,node4,rcWall_SupSection_SectionObj);
        % Add the Elements that define the element
        modelObj.addElement(quad4NObjects {superiorElements(i)});
    end % for i

    % We add the Boundary Elements

```

```

numberElements_Area = length(boundaryElements);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements(i),2);
    node2 = elements2D(boundaryElements(i),3);
    node3 = elements2D(boundaryElements(i),4);
    node4 = elements2D(boundaryElements(i),5);
    quad4NObjects {boundaryElements(i)} = Quad4N(['E'
num2str(boundaryElements(i))],node1,node2,node3,node4,rcWall_AreaConf_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements(i)});
end % for i

% We add the Boundary Elements 2
numberElements_Area = length(boundaryElements2);
for i = 1:numberElements_Area
    node1 = elements2D(boundaryElements2(i),2);
    node2 = elements2D(boundaryElements2(i),3);
    node3 = elements2D(boundaryElements2(i),4);
    node4 = elements2D(boundaryElements2(i),5);
    quad4NObjects {boundaryElements2(i)} = Quad4N(['E'
num2str(boundaryElements2(i))],node1,node2,node3,node4,rcWall_AreaConf2_SectionObj,optionT
ypeElement);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {boundaryElements2(i)});
end % for i

% We add the Center Elements
numberElements = length(centerElements);
for i = 1:numberElements
    node1 = elements2D(centerElements(i),2);
    node2 = elements2D(centerElements(i),3);
    node3 = elements2D(centerElements(i),4);
    node4 = elements2D(centerElements(i),5);
    quad4NObjects {centerElements(i)} = Quad4N(['E'
num2str(centerElements(i))],node1,node2,node3,node4,rcWall_Area_SectionObj);
    % Add the Elements that define the element
    modelObj.addElement(quad4NObjects {centerElements(i)});
end % for i

end % if optionTypeElement
numberElements = size(elements2D,1);

%% Create the Constraint
numberConstraints = length(nodesSPC);
spcObjects = cell(numberConstraints,1);
for i=1:numberConstraints
    spcObjects {i} = FixRestraint(['SPC_N' num2str(nodesSPC(i))],nodesSPC(i),[1 2 3]);
    % We add the SP_Constraint

```

```

    modelObj.addSP_Constraint(spcObjects {i});
end % for i

```

```

%% We store the time to take create the Model
timeModel = toc;
fprintf("Time to initialize the Model Object and ModelComponents Objects is %s [sec]
\n',num2str(timeModel));

```

```

%% We proceed to set The type of Nodes Reactions to Calculate
typeReactions = 0;
flagNodesReactions = true;
modelObj.setTypeNodesReactions(typeReactions,flagNodesReactions);

```

Generador de Mallado

```

Mesh_fc2(1,1910,4470,12,26,90,100,400,'Nodes_R2_RW.txt','Elements_R2_RW.txt')

```

Gráfico de Mallado

```

%% Define Variables Depending of the Model
NameAxesModel = {'Modelacion de Muro - R2'};
sizeSimbolSPC = 40;

```

```

%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

```

```

% We create the AxesModelHandler

```

```

fg{6}=figure();
axesModelHandlerObjects {1} = AxesModelHandler(NameAxesModel {1},fg {6});%,[.05 .4 .4]);
%'Position',[ 1025 545 1100*1.3 850*1.3 ]

```

```

% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {1});

```

```

% We set the LoadPattern to Draw in the AxesModel 1
shlModelGraphicHandlerObj.setLoadPatternToDraw(NameAxesModel {1},shlLoadPatternTag,'Undeformed');
shlModelGraphicHandlerObj.drawElements(axesModelHandlerObjects {1});
shlModelGraphicHandlerObj.drawNodes(axesModelHandlerObjects {1});
%shlModelGraphicHandlerObj.drawNodeTags(axesModelHandlerObjects {1});
shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {1},sizeSimbolSPC);
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementShadow')
shlModelGraphicHandlerObj.setElementPencilGraphicByComponent('ElementClass')

```

```
timePerformAnalysis = toc;
fprintf('Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));

savefig('FiguraMesh_R2');
```

Gráfico de Tensión y Deformación

```
%% Define Variables Depending of the Model
NameAxesModel
= {'RW2_TW_Undeformed','RW2_TW_Deformed','RW2_TW_Undeformed_CrackPattern'};
newGraphicScaleFactor = 1000;
sizeSimbolSPC = 40;

tic
%% We create the ModelGraphicHandler
shlModelGraphicHandlerObj = ModelGraphicHandler(modelObj);
shlModelGraphicHandlerObj.setModelResponseRecorder(shlModelResponseRecorderObj);

% We create the AxesModelHandler

Gs = cell(1);
axesModelHandlerObjects = cell(6,1);

Gs{1} = figure();
set(Gs{1},'Units','inches','Position',[ 10 6 12.5 14 ],'papersize',[12.5 14],'paperposition',[0 0 12.5
14],'color','w','defaultFigurePaperPositionMode','auto');

tic

axesModelHandlerObjects{1} = AxesModelHandler('RW2_TW_Undeformed_S11',Gs{1},[.05 .5
.3 0.5]);
axesModelHandlerObjects{2} = AxesModelHandler('RW2_TW_Undeformed_S22',Gs{1},[.35 .5
.3 0.5]);
axesModelHandlerObjects{3} = AxesModelHandler('RW2_TW_Undeformed_S12',Gs{1},[.65 .5
.3 0.5]);
axesModelHandlerObjects{4} = AxesModelHandler('RW2_TW_Undeformed_E11',Gs{1},[.05 .0
.3 .5]);
axesModelHandlerObjects{5} = AxesModelHandler('RW2_TW_Undeformed_E22',Gs{1},[.35 .0
.3 .5]);
axesModelHandlerObjects{6} = AxesModelHandler('RW2_TW_Undeformed_E12',Gs{1},[.65 .0
.3 .5]);

setBackgroundcolor(axesModelHandlerObjects{1},'white')
setBackgroundcolor(axesModelHandlerObjects{2},'white')
setBackgroundcolor(axesModelHandlerObjects{3},'white')
setBackgroundcolor(axesModelHandlerObjects{4},'white')
setBackgroundcolor(axesModelHandlerObjects{5},'white')
setBackgroundcolor(axesModelHandlerObjects{6},'white')
```

```

% We add the AxesModelHandler to the ModelGraphicHandler
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {1});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {2});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {3});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {4});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {5});
shlModelGraphicHandlerObj.addAxesModelHandler(axesModelHandlerObjects {6});

% We set the type of ResponseToPlot in each AxesModel
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S11','S11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S22','S22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_S12','S12','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E11','E11','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E22','E22','Undeformed');
shlModelGraphicHandlerObj.setTypeResponseToDraw('RW2_TW_Undeformed_E12','E12','Undeformed');

% We set the Scale of the Deformation to draw
shlModelGraphicHandlerObj.setGraphicDeformationScaleFactor(newGraphicScaleFactor);
% We proceed to draw the Last Number of AnalysisStep

for i= Step2Draw
    % First we set the AnalysisStep to Draw
    shlModelGraphicHandlerObj.setAnalysisStepNumber(i);
    % We draw the Elemets

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {1},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {2},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {3},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {4},true);

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {5},true);

```

```

shlModelGraphicHandlerObj.drawElementsWithAverageResponse(axesModelHandlerObjects {6}, true);
    % We draw the SP_Constraint
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {1},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {2},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {3},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {4},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {5},sizeSimbolSPC);
    shlModelGraphicHandlerObj.drawSP_Constraints(axesModelHandlerObjects {6},sizeSimbolSPC);

end
timePerformAnalysis = toc;
fprintf('Time to perform the Graphic of the Model: %s [sec] \n',num2str(timePerformAnalysis));

```

Ejecutor

```

%We close all the Windows
close all
opt = 0;
if opt==0
    % We clear the memory and the classes and functions
    clear classes;
    clear all;
elseif opt == 1
    % We clear the model, analysis model and modelGraphicHandler
    modelObj.clearAll();
    svlModelResponseRecorderObj.clearAll()
    svlAnalysisModelObj.clearAll()
    shlModelResponseRecorderObj.clearAll()
    shlAnalysisModelObj.clearAll()
    shlModelGraphicHandlerObj.clearAll();
    clear variables
    clear all;
elseif opt == 2
    modelObj.clearAll();
    svlModelResponseRecorderObj.clearAll()
    shlModelResponseRecorderObj.clearAll()
    shlModelGraphicHandlerObj.clearAll();
    clear variables
    clear all;
end
clc

nameFileMat = 'Output_R2_RW2';

% Option to create the Model

% Save or Load Option:

```

```

%           1 = Save
%           2 = Load
optionSaveOrLoad = 2;
% Mesh:
%           1 = Simple Mesh
optionMesh = 1;

% Specimen Wall Option:
%           2 = RW2
optionSpecimenWall = 2;

% Concrete Option:
%           1 =
%           OrthotropicRotatingAngleConcreteTensionStiffeningT2DMaterial01
optionConcrete = 1;

% Steel Option:
%           1 = SmearredEmbeddedSteelDoubleLayerROT2DMaterial01
%           2 = SmearredEmbeddedSteelDoubleLayerMPT2DMaterial01
%           3 = SteelDoubleLayer2DMaterial

optionSteel = 2;

% Damage Option:
%           0 = No Damage
%           1 = Damage
optionDamage = 1;

% Type of Element:
%           1 = Quad4N
%           2 = Quad4NR02
optionTypeElement = 2;

% SelfWeight Option:
%           0 = Not Include
%           1 = Include
optionSelfWeight = 1;

%% We create the Model
R2_RW_Model

%% We Create the Static Vertical Load, the Response Recorder and Perform the Analysis
R2_RW_StaticVerticalLoad

%% We Create the Static Horizontal Load, the Response Recorder and Perform the Analysis
R2_RW_StaticHorizontalLoad

%% We create the Graphic for the Static Horizontal Load

```



```

iNumberIncrementRecorded_SHL = numberTotalIncrRecorded_SHL-1;

% We extract the Displacement of the Top of the Wall
dispCM =
shlNodeRRObjets {nodeTagToMonitorInTheDisplacementControl}.getResponseRecorded('Displ');
% We extract the Total Horizontal Load apply at the Top of the Wall
loadCM = numberHorizontalNodeLoads*shlNodeLoadRRObjets {1}.getResponseRecorded();
%% We Create Variable of Displacement
Wall_displ=zeros(numberTotalIncrRecorded_SHL,1);
Wall_displ=dispCM(:,dofNumberHorizontalLoad {1});
Wall_load=loadCM(:,dofNumberHorizontalLoad {1})/1000;

%% %% Graph Step 1%
one=find(Wall_displ(:,1) > height_Wall/100 & Wall_displ(:,1) < height_Wall/100+1);
Step2Draw = one(1);
R2_RW_GraphicInStep

%% Graph Step Max V
Step2Draw=find(Wall_load==max(Wall_load));
R2_RW_GraphicInStep

%% Graphic Element
R2_RW_ElementGraphic

% We create the Figure that compare the top wall displacement response between the Experiment vs
the Analytical response
fg = figure();
% We plot the graphics
Exp_Results=load('R2_ExperimentalResults.txt');
plot(Exp_Results(:,1),Exp_Results(:,2),'b','LineWidth',0.5);
hold on
plot(Wall_displ(:,1),Wall_load,'Color',[1 0 0],'LineWidth',0.5);
xlabel('Desplazamiento [mm]');
ylabel('Fuerza [kN]');
legend('R2','MR5');
grid on
savefig('FiguraAux_R2');

```

Función Nodos y Elementos

```
function [Nodos,Elementos] =
Mesh_fc2(Tipodemuro,ancho,alto,DivisionH,DivisionV,anchoconf1,anchoconf2,zonaCarga,NodosT
XT,ElementosTXT)

%Tipo de muro:
% 1 = Tipo Doble T
% 2 = Tipo T
% Mesh_fc(...
% Tipodemuro,
% ancho,
% alto,
% N? elem. horizontales,
% N? elem. verticales 1,
% Ancho de confinamiento exterior,
% Ancho de confinamiento interior,
% zona de Carga
% 'Nombre Texto de nodos'
% 'Nombre Texto de elementos'

Archivo.Nodos=fopen(NodosTXT,'w');
Archivo.Elementos=fopen(ElementosTXT,'w');

if Tipodemuro==1

    NNH=DivisionH+1;%Numero de Nodos Horizontales
    NNV=DivisionV+1;%Numero de Nodos Verticales
    NNT=(NNV+1)*(NNH+1)+8*(NNH+1)+8*5;%Nodos Totales
    DeltaH=(ancho-2*anchoconf1-2*anchoconf2)/(DivisionH-3); %separacion horizontal entre
nodos
    DeltaV=(alto)/(NNV);%Separacion vertical entre nodos
    DeltaV_Sup=zonaCarga/4;
    DivisionV=DivisionV+1;

    %Creacion de variables

    Nodos=zeros(NNT,3);
    Elementos=zeros(4*(DivisionH+1+4)+(DivisionH+1)*(DivisionV)+4*(DivisionH+1+4),5);
    length(Elementos(:,1))
    h=0;
    v=1;
    sv=1;

    n=1;
    k=1;
    q=1;
```

```

%Nodos
for i=1:NNI
    Nodos(i,1)=i;
    if i<=(20+5*(NNH+1))
        if h==0
            Nodos(i,2)=0;
        elseif h==3 || h==2+(NNH)
            Nodos(i,2)=Nodos(i-1,2)+anchoconf1;
        elseif h==4 || h==1+(NNH)
            Nodos(i,2)=Nodos(i-1,2)+anchoconf2;
        else
            Nodos(i,2)=Nodos(i-1,2)+DeltaH;
        end
        Nodos(i,3)=DeltaV*(v-1);
        if h==(NNH+4)
            h=0;
            if v==5
                v=1;
            else
                v=v+1;
            end
        else
            h=h+1;
        end
    elseif i>(20+5*(NNH+1)+(NNH+1)*(NNV-1))
        if h==0
            Nodos(i,2)=0;
        elseif h==3 || h==2+(NNH)
            Nodos(i,2)=Nodos(i-1,2)+anchoconf1;
        elseif h==4 || h==1+(NNH)
            Nodos(i,2)=Nodos(i-1,2)+anchoconf2;
        else
            Nodos(i,2)=Nodos(i-1,2)+DeltaH;
        end
        Nodos(i,3)=DeltaV*(v+4)+DeltaV_Sup*(sv-1);
        if h==(NNH+4)
            h=0;
            if sv==5
                sv=1;
            else
                sv=sv+1;
            end
        else
            h=h+1;
        end
    end
elseif i>(20+5*(NNH+1)) && i<=(20+5*(NNH+1)+(NNH+1)*(NNV-1))
    if h==0
        Nodos(i,2)=2*DeltaH;
    end
end
    
```

```

elseif h==1 || h==NNH
    Nodos(i,2)=Nodos(i-1,2)+anchoconf1;
elseif h==2 || h==NNH-1
    Nodos(i,2)=Nodos(i-1,2)+anchoconf2;
else
    Nodos(i,2)=Nodos(i-1,2)+DeltaH;
end

Nodos(i,3)=DeltaV*(v+4);

if h==NNH
    h=0;
    v=v+1;
else
    h=h+1;
end
end
end

%Elementos
for i=1:length(Elementos(:,1))
    Elementos(i,1)=i;
    if i<=(4*(4+DivisionH+1))
        Elementos(i,2)=n;
        Elementos(i,3)=n+1;
        Elementos(i,4)=n+(4+NNH+1)+1;
        Elementos(i,5)=n+(4+NNH+1);
        if k==4+DivisionH+1
            k=1;
            if q==4
                q=1;
                n=n+4;
            else
                q=q+1;
                n=n+2;
            end
        else
            k=k+1;
            n=n+1;
        end
    end

elseif i>(4*(4+DivisionH+1)+(DivisionH+1)*(DivisionV))
    Elementos(i,2)=n;
    Elementos(i,3)=n+1;
    Elementos(i,4)=n+(4+NNH+1)+1;
    Elementos(i,5)=n+(4+NNH+1);
    if k==4+DivisionH+1
        k=1;
        q=q+1;
    end
end

```

```

        n=n+2;
    else
        k=k+1;
        n=n+1;
    end
elseif i>(4*(4+DivisionH+1)) && i<=(4*(4+DivisionH+1))+(DivisionH+1)*(DivisionV)
    if q==1
        Elementos(i,2)=n;
        Elementos(i,3)=n+1;
        Elementos(i,4)=n+NNH+4;
        Elementos(i,5)=n+NNH+3;
        if k==DivisionH+1
            n=n+4;
            k=1;
            q=q+1;
        else
            n=n+1;
            k=k+1;
        end
    elseif q==DivisionV
        Elementos(i,2)=n;
        Elementos(i,3)=n+1;
        Elementos(i,4)=n+NNH+4;
        Elementos(i,5)=n+NNH+3;
        if k~=DivisionH+1
            n=n+1;
            k=k+1;
        elseif k==DivisionH+1
            k=1;
            q=1;
            n=n+2;
        end
    else
        Elementos(i,2)=n;
        Elementos(i,3)=n+1;
        Elementos(i,4)=n+(NNH+1)+1;
        Elementos(i,5)=n+(NNH+1);
        if k==DivisionH+1
            k=1;
            q=q+1;
            n=n+2;
        else
            k=k+1;
            n=n+1;
        end
    end
end
end
end
end

```

```

elseif Tipodemuro==2

    NNH=DivisionH+1;%Numero de Nodos Horizontales
    NNV=DivisionV+1;%Numero de Nodos Verticales
    NNT=(NNV+1)*(NNH+1)+8*(NNH+1)+4*5;%Nodos Totales
    DeltaH=(ancho-2*anchoconf1-2*anchoconf2)/(DivisionH-3); %separacion horizontal entre
nodos
    DeltaV=(alto)/(NNV);%Separacion vertical entre nodos
    DivisionV=DivisionV+1;
    DeltaV_Sup=zonaCarga/4;

    %Creacion de variables

    Nodos=zeros(NNT,3);
    Elementos=zeros(4*(DivisionH+1+4)+(DivisionH+1)*(DivisionV)+4*(DivisionH+1),5);
    length(Elementos(:,1))
    h=0;
    v=1;

    n=1;
    k=1;
    q=1;

    %Nodos
    for i=1:NNT
        Nodos(i,1)=i;
        if i<=(20+5*(NNH+1))
            if h==0
                Nodos(i,2)=0;
            elseif h==3 || h==2+(NNH)
                Nodos(i,2)=Nodos(i-1,2)+anchoconf1;
            elseif h==4 || h==1+(NNH)
                Nodos(i,2)=Nodos(i-1,2)+anchoconf2;
            else
                Nodos(i,2)=Nodos(i-1,2)+DeltaH;
            end
            Nodos(i,3)=DeltaV*(v-1);
            if h==(NNH+4)
                h=0;
                if v==5
                    v=1;
                else
                    v=v+1;
                end
            else
                h=h+1;
            end
        else
            if h==0

```

```

        Nodos(i,2)=2*DeltaH;
    elseif h==1 || h==NNH
        Nodos(i,2)=Nodos(i-1,2)+anchoconf1;
    elseif h==2 || h==NNH-1
        Nodos(i,2)=Nodos(i-1,2)+anchoconf2;
    else
        Nodos(i,2)=Nodos(i-1,2)+DeltaH;
    end

    if v>=NNV
        Nodos(i,3)=DeltaV*(NNV+4)+DeltaV_Sup*(v-NNV);
    else
        Nodos(i,3)=DeltaV*(v+4);
    end

    if h==NNH
        h=0;
        v=v+1;
    else
        h=h+1;
    end
end
end

%Elementos
for i=1:length(Elementos(:,1))
    Elementos(i,1)=i;
    if i<=(4*(4+DivisionH+1))
        Elementos(i,2)=n;
        Elementos(i,3)=n+1;
        Elementos(i,4)=n+(4+NNH+1)+1;
        Elementos(i,5)=n+(4+NNH+1);
        if k==4+DivisionH+1
            k=1;
            if q==4
                q=1;
                n=n+4;
            else
                q=q+1;
                n=n+2;
            end
        else
            k=k+1;
            n=n+1;
        end
    else
        if q==1
            Elementos(i,2)=n;
            Elementos(i,3)=n+1;

```

```

        Elementos(i,4)=n+NNH+4;
        Elementos(i,5)=n+NNH+3;
        if k==DivisionH+1
            n=n+4;
            k=1;
            q=q+1;
        else
            n=n+1;
            k=k+1;
        end
    end
else
    Elementos(i,2)=n;
    Elementos(i,3)=n+1;
    Elementos(i,4)=n+(NNH+1)+1;
    Elementos(i,5)=n+(NNH+1);
    if k==DivisionH+1
        k=1;
        q=q+1;
        n=n+2;
    else
        k=k+1;
        n=n+1;
    end
end
end
end
end
end

fprintf(Archivo.Nodos,'%1.0i %1.3f %1.3f\r\n',Nodos);
fprintf(Archivo.Elementos,'%5d %5d %5d %5d %5d\n',Elementos);
fclose(Archivo.Elementos);
fclose(Archivo.Nodos);

scatter(Nodos(:,2),Nodos(:,3))

end

```