



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Formulación e implantación
de métodos de acoplamiento
orientados al problema de
interacción suelo-estructura**

TESIS

Que para obtener el título de

Ingeniera Civil

P R E S E N T A

Daniela Bañuelos Gutiérrez

DIRECTOR DE TESIS

Dr. Leonardo Ramírez Guzmán



Ciudad Universitaria, Cd. Mx., 2026



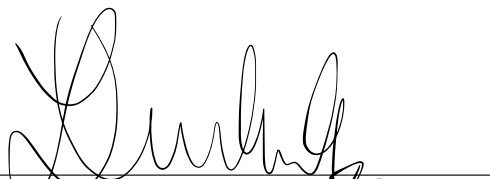
**PROTESTA UNIVERSITARIA DE INTEGRIDAD Y
HONESTIDAD ACADÉMICA Y PROFESIONAL
(Titulación con trabajo escrito)**



De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado FORMULACION E IMPLANTACION DE METODOS DE ACOPLAMIENTO ORIENTADOS AL PROBLEMA DE INTERACCION SUELO-ESTRUCTURA que presenté para obtener el título de INGENIERA CIVIL es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.


DANIÉLA BAÑUELOS GUTIERREZ
Número de cuenta: 316316488

Agradecimientos

A MI MAMÁ. Ser tu hija es el honor más grande que he podido tener. Todo lo que fui, soy y seré es por ti; me enseñaste a pensar. Convertiste tus sacrificios en mis oportunidades. Gracias por actuar como si ya estuviera destinada a llegar lejos. Creíste en mí, mamá. Lo logramos.

A MI FAMILIA por su apoyo absolutamente incondicional ante las adversidades. A mi abuelito Jesús, por fomentar en mí hambre de conocimiento. Gran parte de este trabajo existe porque nunca me dejabas de contar historias sobre el mundo. Te extraño todos los días. A mi abuelita Teresa por cuidar de mí como una protectora absolutamente incondicional: tu amor llena de alegría mis días. A mi tía Sara Elia por entender un corazón frágil y una mente sensible. A mi tío Arturo por ser referencia absoluta de entrega a las personas que lo rodean: gracias por tratarme como una hija más. Y a mi primo Arturo, por ser fuente infinita de inspiración humana, académica y personal; me recuerdas que siempre llegan tiempos mejores.

AL INSTITUTO DE INGENIERÍA Y AL DR. LEONARDO RAMÍREZ GUZMÁN, por su apoyo incondicional como asesor y ejemplo de ser humano. Por su escucha activa; por ser un gran consejero académico y personal. Muchas de sus enseñanzas y consejos forman ahora parte de mi manera de entender la ingeniería, la investigación y la vida. Gracias por la oportunidad todos estos años. A mis compañeras y compañeros de la Unidad de Instrumentación Sísmica que con su paciencia y su conocimiento hicieron el proceso no solo más ligero, sino amable. Gracias Fernando, Héctor, Iván, Lilibeth, Luis, Miguel y Paula.

AL COMITÉ DE ESTE TRABAJO por tratar mis ideas como algo digno de discusión. Ustedes dejaron una huella profunda en mi formación académica y humana. Sus clases, observaciones y preguntas han moldeado mi manera de pensar. Gracias por compartir generosamente su conocimiento, su experiencia y las historias que los llevaron a convertirse en los investigadores, ingenieros y consultores que son hoy. Son mi ejemplo de rigor intelectual, curiosidad y compromiso con el conocimiento. Sus enseñanzas permanecerán conmigo mucho después de estas páginas. Gracias al Dr. Jaime de Jesús Paredes Camacho, al Mtro. José Antonio Pérez Pontón, al M.I. Luis Ignacio Román de la Sancha y al Dr. Raúl Sánchez García.

A MIS AMIGAS Y AMIGOS que me brindaron su amistad, su tiempo, sus palabras y sus consejos. Gracias por proveerme una red de apoyo y confianza, por la motivación que me dieron en cada semestre: gracias por hacerme sentir segura y protegida. Por las aventuras; por los días y las noches en las que me hicieron compañía. Su amistad fue contención a lo largo de estos años. Gracias Abraham, Alejandro, Ana Karla, Ana Sofía, Andrés, Eduardo, Eliezer, Francisco, Leonardo, Luis Roberto, Marco, Mauricio, Michelle, Omar, Rafael, Sara, Uriel, Uziel y Venus. Su amistad, compañía y consuelo incondicional son fuente de mi resiliencia. Su vida inspira la mía.

ABSTRACT

Mexico City's Zone III lacustrine soils amplify seismic demand in ways that fixed-base models cannot capture. This thesis implements a soil-structure coupling scheme in MATLAB using equivalent dynamic impedances from Gazetas as boundary conditions at base nodes, replacing the conventional fixed-base assumption. The model was validated against SAP2000 with good agreement in modal periods and displacements.

Soil-structure interaction was evaluated using the superposition method of Avilés López et al. The fundamental period elongates relative to the fixed-base condition, with rocking as the dominant contributor to that shift. The result falls below the range commonly reported for lacustrine deposits, but is physically consistent with the foundation plan dimensions, which produce high lateral stiffness. The proposed scheme offers a viable computational tool for preliminary seismic assessment of buildings on soft soil, within the normative framework of NTC-Sismo and NTC-Cimentaciones.

RESUMEN EJECUTIVO

Esta tesis implementa un esquema de acoplamiento suelo-estructura en MATLAB, dentro del rango elástico lineal, usando impedancias dinámicas equivalentes de Gazetas como condiciones de frontera en los nodos de la base en lugar del empotramiento convencional. El modelo se aplicó a un marco tridimensional de concreto reforzado desplantado en la Zona III de Lago de la Ciudad de México y se validó contra SAP2000 con buena concordancia en periodos y desplazamientos.

La interacción suelo-estructura se analizó con el método de superposición de Avilés López et al. El periodo fundamental se alarga respecto a la condición de base rígida y la mayor parte de ese incremento proviene del cabeceo de la cimentación. El resultado es menor al rango típico para suelos lacustres, pero tiene sentido en el modelo físico. Las dimensiones en planta de la cimentación producen alta rigidez lateral. El esquema propuesto funciona como herramienta de cálculo para evaluación sísmica preliminar en suelo blando, dentro del marco de las NTC-Sismo y NTC-Cimentaciones.

Índice general

INTRODUCCIÓN	I
o.1. Planteamiento del Problema	1
o.2. Hipótesis	2
o.3. Objetivos	3
o.4. Metodología de la investigación	4
o.5. Marco conceptual	5
o.6. Marco teórico	10
o.7. Alcance y delimitaciones del estudio	13
1. MÉTODOS DE ANÁLISIS ESTRUCTURAL	15
1.1. Características de la sismicidad en regiones de Alta Vulnerabilidad Estructural	15
1.2. Influencia de la sismicidad en el Diseño Estructural	18
1.3. Métodos de Análisis Estructural en Zonas Sísmicas	20
1.4. Espectros de Diseño Sísmico	24
1.5. Análisis Estructural basado en la Matriz de Rigideces	26
2. IMPLEMENTACIÓN DEL CÓDIGO PARA EL ANÁLISIS SÍSMICO	37
2.1. Descripción de las funciones del código en MATLAB	37
2.2. Desarrollo del Código	53
3. ANÁLISIS DE RESULTADOS	108
3.1. Modelo de Validación en SAP2000	109
3.2. Comparación Cuantitativa de los Resultados obtenidos	113
3.3. Análisis de diferencias en comportamiento estructural	121
3.4. Limitaciones del Código Implementado	127
3.5. Interpretación de los Resultados y Fiabilidad del Modelo	132
4. CONCLUSIONES	133
BIBLIOGRAFÍA	136
APÉNDICE A. RESULTADOS DE MATLAB	141
A.1. Resultados de generación de la geometría de la estructura	141
A.2. Resultados de generación de la Matriz de Rigidez	143
A.3. Resultados del Análisis Estático	145
A.4. Resultados del Análisis Dinámico	148

A.5. Resultados de Análisis de Interacción Suelo-Estructura	151
APÉNDICE B. CÓDIGOS DE ANÁLISIS [MATLAB R2024B]	159
B.1. marco3d_editable.m	159
B.2. mrigidez.m	176
B.3. analisis_estatico.m	193
B.4. analisis_dinamico.m	207
B.5. analisis_ise_superposicion.m	226

Índice de figuras

1.	Esquema del fenómeno en interacción cinemática e inercial (Avilés López et al., 2016).	11
1.1.	Comparación entre los periodos predominantes calculados mediante relaciones R_{ij} y cocientes espectrales H/V , frente a los valores asignados por el mapa de zonificación sísmica del RCCDMX. Adaptado de Diego Melgar y Çelebi, 2018	19
1.2.	Ejemplo 1.A	28
2.1.	Diagrama General de decisiones de los scripts del código. Continúa en la siguiente página.	39
2.2.	Continuación del Diagrama General de los scripts del código.	40
2.3.	Distribución de masas nodales traslacionales del modelo estructural, resultado del proceso de tributación de peso propio según columnas, vigas y losas. El color está relacionado a la magnitud de la masa concentrada en cada nodo.	42
2.4.	Marco en dos direcciones con vigas, columnas y losas. Incluye nodos numerados.	43
2.5.	Patrón de banda de la matriz de rigidez global K tras la aplicación del diafragma rígido. La reducción de grados de libertad refleja la condensación cinemática de los desplazamientos horizontales de cada nivel en los grados de libertad maestros.	48
2.6.	Patrón de banda de la matriz de rigidez reducida K_{red} tras la eliminación de los grados de libertad restringidos en la base. Los grados de libertad libres resultantes son los que participan en los análisis dinámicos subsecuentes.	49
2.7.	Diagrama de decisiones del análisis estático. Continúa en la siguiente página.	55
2.8.	Continuación del diagrama de decisiones del Análisis Estático.	56
2.9.	Vista tridimensional de la deformada bajo carga puntual estática. La imagen superior corresponde a escala real ($1\times$) y la inferior a una amplificación de $500\times$, únicamente con fines visuales.	63
2.10.	Diagrama del Análisis Dinámico que incluye análisis modal y espectral. Continúa en la siguiente página.	68
2.11.	Continuación del diagrama de decisiones del Análisis Dinámico.	69
2.12.	Espectro de aceleraciones obtenido del SASID para suelo Tipo III - Lago. Los puntos negros indican los primeros seis modos de vibrar del edificio.	74
2.13.	Deformada espectral bajo excitación en un eje. La estructura original se representa en negro y la deformada en rojo punteado.	77
2.14.	Deformada espectral bajo excitación en un eje con un factor de amplificación de $10\times$. La estructura original se representa en negro y la deformada en rojo punteado.	78
2.15.	Acelerograma del sismo del 19 de septiembre del 2017. Registrado en la estación SCT2 (Instituto de Geofísica, UNAM, 2017).	80
2.16.	Historia de desplazamientos tiempo-historia en la estructura, provocados por el sismo del 19 de septiembre del 2017.	82

2.17.	Fotograma representativo de la animación tridimensional generada por el código durante el análisis de historia de tiempo.	83
2.18.	Diagrama de proceso del análisis de Interacción Suelo-Estructura. Continúa en la siguiente página.	91
2.19.	Continuación del diagrama de proceso del análisis de Interacción Suelo-Estructura. Continúa en la siguiente página.	92
2.20.	Continuación del diagrama de proceso del análisis de Interacción Suelo-Estructura.	93
2.21.	Comparación de períodos para mostrar efectos de Interacción Suelo-Estructura en los primeros modos de vibrar.	101
3.1.	Vista tridimensional del modelo de base rígida en SAP2000. Las columnas se representan en amarillo y las vigas en azul. Los símbolos verdes en la base corresponden a los empotramientos que restringen los seis grados de libertad de cada nodo a nivel $z = 0$	109
3.2.	Planta del modelo en el plano $X-Y$ a $Z = 42$ m (nivel de azotea). Los cinco ejes en X (A-E) y los cinco en Y (1-5) definen la retícula ortogonal con claros de 5 m y 6 m respectivamente.	110
3.3.	Marco en el plano $X-Z$ a $Y = 24$ m. La distribución regular de elementos en los 12 niveles de 3.5 m de altura confirma la ausencia de irregularidades en elevación. Los cuadros verdes en la base son los apoyos de empotramiento.	111
3.4.	Detalle de la zona de cimentación del modelo en SAP2000 con renderizado de secciones. Las secciones de columnas (50 × 50 cm) y vigas (50 × 30 cm) son visibles en la representación tridimensional. Los cuadros verdes en cada nodo de la base corresponden a los empotramientos perfectos.	112
3.5.	Cargas distribuidas de losa (caso LOSA) aplicadas sobre las vigas del modelo en SAP2000. La magnitud uniforme de 12.84 kN/m en todos los niveles es consistente con la carga superficial de 600 kg/m ² distribuida según el área tributaria de cada viga.	113
3.6.	Forma modal del modo 1 en SAP2000, base rígida. Período $T_1 = 2.423$ s, frecuencia $f_1 = 0.413$ Hz. La deformada muestra traslación predominante en dirección Y con amplitud máxima en la azotea.	115
3.7.	Forma modal del modo 1 en SAP2000, base rígida, vista en el plano $Y-Z$. Se aprecia el perfil de desplazamientos laterales tipo ménsula con curvatura creciente hacia los niveles superiores, característico del modo fundamental de traslación.	116
3.8.	Deformada del modelo bajo carga estática lateral en dirección Y (caso EST_RIGIDEZ_Y) en SAP2000. La forma de la deformada es consistente con la distribución de rigidez lateral del sistema de marcos en esa dirección.	119
3.9.	Estado inicial de la estructura en el análisis tiempo-historia en dirección Y (caso TH_Y, $t = 0$ s) en SAP2000. La estructura se encuentra en reposo antes de la aplicación del acelerograma SCT2-2017.	120
3.10.	Forma modal del modo 1 en el modelo ISE de SAP2000, vista tridimensional frontal. Período $\tilde{T}_1 = 2.786$ s, frecuencia $\tilde{f}_1 = 0.359$ Hz. La deformada incorpora traslación de la base y rotación por cabeceo, que no están presentes en el modelo de base rígida.	122
3.11.	Forma modal del modo 1 en el modelo ISE de SAP2000, vista tridimensional lateral. Se observa la inclinación de la base respecto a la vertical, que refleja la deformación rotacional del resorte de cabeceo K_r	123

3.12.	Forma modal del modo 1 en el modelo ISE de SAP2000, vista en el plano $Y-Z$. La rotación en la base es visible en la inclinación de las columnas del primer nivel; el perfil de desplazamientos resultante combina la deformación de la superestructura con la traslación y el cabeceo de la cimentación.	124
3.13.	Vista tridimensional del modelo ISE en SAP2000. La ausencia de símbolos de empotramiento en la base distingue este modelo del de base rígida; en su lugar se asignan resortes lineales equivalentes en el nodo central de la cimentación.	129
3.14.	Detalle de la zona de cimentación del modelo ISE en SAP2000. Las columnas descansan sobre nodos libres en $z = 0$, conectados al nodo central mediante vínculos rígidos que concentran la flexibilidad del suelo en un único punto de aplicación de los resortes equivalentes.	130
3.15.	Planta del modelo ISE en SAP2000 a nivel $z = 0$. El nodo central (marcado en verde) concentra las rigideces equivalentes del suelo K_b y K_r ; los nodos perimetrales de la base están vinculados a él mediante restricciones rígidas que reproducen el comportamiento de la losa de cimentación como cuerpo rígido.	131

Introducción

0.1. PLANTEAMIENTO DEL PROBLEMA

Debido a que las ondas sísmicas generan efectos distintos según la zona a la que arriban, el Reglamento de Construcciones de la CDMX estableció una zonificación geotécnica que divide al territorio en: Zona I (Lomas), Zona II (Transición) y Zona III (Lacustre). En los depósitos blandos de los antiguos lagos, las ondas se amplifican considerablemente; este efecto de sitio puede transformar un movimiento moderado en uno de alta demanda estructural para edificios ubicados en la zona lacustre, donde los daños observados en sismos recientes se atribuyen precisamente a este mecanismo en suelos arcillosos. La magnitud o la distancia al epicentro no constituyen los factores más determinantes de la intensidad sísmica en la Ciudad de México; el factor crítico es la amplificación local producida por las propiedades dinámicas del depósito de arcillas del Valle de México, que puede incrementar las aceleraciones espectrales hasta en un factor de 5 respecto al basamento de roca (Celebi et al., 1987).

En eventos pasados, como el del 19 de septiembre de 2017, se registraron aceleraciones hasta cinco veces mayores en la Zona III que en la Zona I (Zona de Lomas) durante el mismo evento, lo que evidencia la necesidad de metodologías de diseño estructural que consideren explícitamente la respuesta dinámica en suelos blandos. También es importante considerar que los peligros provocados por los eventos sísmicos no se limitan a aquellos originados a más de 300 km de la capital del país. La presencia de fallas activas en la Zona Metropolitana del Valle de México ha mostrado que la Ciudad está expuesta también a eventos de

origen local (Celebi et al., 1987).

Un recurso ampliamente utilizado en los últimos años han sido las simulaciones computacionales para analizar la interacción de las ondas sísmicas con la estratigrafía local. Sin embargo, muchas de estas simulaciones asumen condiciones ideales en las estructuras o en los movimientos de la base. La interacción entre el suelo y la estructura es bidireccional: la vibración de cada edificio modifica el entorno del suelo en el que la onda se expande, llegando a reflejar, refractar, curvar y dispersar la energía, modificando a la vez la deformación de las estructuras. Este comportamiento, que varía con la geometría y las propiedades de cada estructura, puede incrementar la demanda sísmica de forma significativa en zonas de suelo blando.

El subsuelo lacustre de la Ciudad de México modifica la frecuencia, la amplitud y la duración del movimiento sísmico de manera que los espectros de diseño convencionales no capturan. Ignorar esta condición produce estimaciones de demanda sísmica que no son conservadoras. En zonas lacustres, la frecuencia natural de edificios de mediana y gran altura coincide con la frecuencia predominante de los movimientos sísmicos (Celebi et al., 1987), lo que provoca resonancia y amplificación del movimiento (J. Mayoral et al., 2019). Adicionalmente, la duración del movimiento se extiende en la Zona III respecto a suelo firme, incrementando los ciclos de carga y descarga durante un evento sísmico y, con ello, la acumulación de daño y el riesgo de colapso por fatiga (Quinde y Reinoso, 2020).

Por estas razones, la Zona Metropolitana del Valle de México requiere modelos que incorporen la respuesta dinámica de los edificios, sus interacciones y sus acoplamientos con el suelo. La omisión de este acoplamiento produce estructuras que, aunque calculadas conforme al Reglamento de Construcciones de la Ciudad de México y las Normas Técnicas Complementarias, subestiman la demanda real al no representar la flexibilidad del subsuelo que las sostiene ni el alargamiento del período fundamental asociado a dicha flexibilidad.

0.2. HIPÓTESIS

Se modelará la Interacción Suelo-Estructura mediante rigideces equivalentes traslacionales y rotacionales asignadas en los nodos de la base de la estructura. Estas rigideces representan la flexibilidad dinámica del suelo a través de resortes equivalentes cuyas constantes se calculan conforme a las expresiones de impedancia

de Gazetas, permitiendo capturar el acoplamiento dinámico sin discretizar el medio continuo del suelo.

Esta aplicación está fundamentada en el Método de la Matriz de Rigidez dentro del marco del Método de Elementos Finitos, implementada en MATLAB, lo que permite mejorar la precisión en el cálculo de las reacciones que se presentan en la base de las estructuras durante los eventos sísmicos. La compatibilidad de desplazamientos en la interfaz suelo-estructura se garantiza mediante la asignación de rigideces equivalentes en los nodos de la base, calculadas conforme a las expresiones de Gazetas para cimentación superficial sobre semiespacio elástico. Este esquema permite representar los efectos dinámicos característicos de suelos blandos, como el alargamiento del período fundamental y el incremento del amortiguamiento por radiación, sin necesidad de discretizar el medio continuo del suelo.

Esta metodología incorpora condiciones de frontera compatibles con la compresibilidad del suelo, a diferencia de los modelos convencionales con empotramientos rígidos, lo que permite cuantificar el alargamiento del período estructural producido por la rigidez del suelo, efecto que en zonas lacustres puede modificar de forma significativa la demanda sísmica sobre la estructura.

El modelo se valida mediante comparación con resultados del software comercial SAP2000, empleando la misma geometría, condiciones de frontera, materiales y cargas en ambas plataformas.

0.3. OBJETIVOS

0.3.1. OBJETIVO GENERAL

Desarrollar y documentar un esquema de acoplamiento suelo-estructura mediante impedancias dinámicas equivalentes en los nodos de base dentro del Método de Elementos Finitos en MATLAB para evaluar su impacto en la respuesta dinámica y en los indicadores de demanda estructural bajo registros sísmicos representativos en la Ciudad de México.

0.3.2. OBJETIVOS ESPECÍFICOS

1. Implementar en MATLAB un esquema de acoplamiento suelo–estructura mediante rigideces equivalentes en los nodos de apoyo, dentro del marco del Método de Elementos Finitos (MEF), considerando condiciones realistas de interacción suelo-estructura.
2. Validar el modelo numérico propuesto contra una referencia de software comercial (SAP2000) de análisis estructural, mediante métricas cuantitativas.

0.4. METODOLOGÍA DE LA INVESTIGACIÓN

Esta investigación adopta un enfoque numérico basado en la dinámica estructural matricial, implementada mediante modelos tridimensionales desarrollados en MATLAB. Se analizará la influencia del acoplamiento suelo-estructura sobre la respuesta dinámica de marcos espaciales representativos de edificaciones en la Ciudad de México, considerando la rigidez característica de los suelos blandos de la zona lacustre mediante un esquema de acoplamiento nodal basado en la aplicación de resortes y amortiguadores equivalentes. MATLAB fue seleccionado como plataforma de desarrollo porque permite el control directo sobre cada etapa del ensamblaje matricial, desde la definición de grados de libertad hasta la aplicación de condiciones de frontera, sin depender de rutinas predefinidas. Esta característica es indispensable para implementar el esquema de acoplamiento dinámico mediante rigideces equivalentes, que requiere intervención explícita en los nodos de base y en la modificación del vector de fuerzas. A diferencia de los programas comerciales de análisis estructural, donde el acceso a la formulación interna es limitado, MATLAB permite verificar y modificar directamente el sistema de ecuaciones en cada paso del cálculo. El desarrollo de modelos con el Método de los Elementos Finitos (MEF) exige implementar operaciones matriciales de gran tamaño, como el cálculo de deformaciones y la aplicación directa de las condiciones de frontera, operaciones que MATLAB maneja de forma nativa con control total sobre la formulación.

El entorno de programación elegido permite formular desde cero los sistemas de ecuaciones estructurales e intervenir en cada etapa del proceso de cálculo. Con esta formulación se construirán modelos tridimensio-

nales que simulan el comportamiento dinámico de estructuras con sus propiedades geométricas, estáticas y dinámicas. Se utilizará la configuración espacial de marcos en dos direcciones mediante el ensamblaje de elementos tipo barra y nodos, considerando múltiples niveles y crujiás, representativas de edificaciones en la Zona Metropolitana del Valle de México.

Cada nodo del modelo cuenta con seis grados de libertad, lo que permite representar los efectos de la flexión, torsión y deformación axial. Los elementos estructurales son tipo barra; sus matrices de rigidez locales se ensamblan para conformar la matriz global del sistema. Se desarrollarán tres tipos de análisis: estático, para verificación de equilibrio y validación de cargas gravitacionales; dinámico, orientado a la obtención de frecuencias naturales y modos de vibración; y análisis de interacción suelo-estructura.

El suelo se modelará mediante apoyos empotrados tridimensionales idealizados, sin discretización del medio continuo, lo que permite reducir el tiempo de procesamiento computacional. Sobre estos apoyos se introduce el acoplamiento dinámico mediante resortes traslacionales y rotacionales entre los nodos base de la estructura y nodos que simulan la reacción del suelo.

La validez del modelo desarrollado en MATLAB se verificará mediante comparación con modelos equivalentes en SAP2000. Se replicarán la geometría, condiciones de frontera, materiales y cargas, y se compararán los resultados obtenidos: desplazamientos máximos, frecuencias naturales y esfuerzos internos. Además, se analizará una versión del modelo con apoyos rígidos, lo que permitirá cuantificar el efecto específico del acoplamiento sobre la respuesta estructural. Mientras que en SAP2000 se modelan apoyos rígidos, en MATLAB el acoplamiento se representa mediante rigideces equivalentes aplicadas en los nodos empotrados de la base.

0.5. MARCO CONCEPTUAL

Este trabajo desarrolla un modelo numérico para estudiar la Interacción Suelo-Estructura (ISE) en condiciones dinámicas, con énfasis en edificaciones localizadas sobre los suelos blandos de la Zona Metropolitana del Valle de México. A continuación se presentan los fundamentos teóricos del acoplamiento e interacción suelo-estructura que sustentan el modelo implementado.

Los Grados de Libertad (GL) son los componentes de desplazamiento o las coordenadas necesarias para

definir la posición de un sistema en el espacio en cualquier momento bajo cualquier carga. En un entorno cotidiano, las estructuras pueden tener un número infinito de grados de libertad, ya que cada elemento puede deformarse de distintas maneras (McGuire et al., 2000). Sin embargo, para el análisis y diseño está permitido evitar esa complejidad a través de la discretización estableciendo un mínimo de grados de libertad (GL) para obtener un resultado adecuado. Por ejemplo, en el análisis dinámico se utiliza un número finito de GL al concentrar masas en puntos previamente establecidos.

Debido a que no todas las deformaciones tienen la misma magnitud, algunos Grados de Libertad pueden ser desestimados sin perder demasiada precisión (McGuire et al., 2000). Esto se observa, por ejemplo, en un apoyo empotrado en 3D. Este apoyo restringe los seis grados de libertad de un nodo a tres desplazamientos (u_x, u_y, u_z) y tres rotaciones o giros: $(\theta_x, \theta_y, \theta_z)$. Se utiliza en este trabajo como condición de frontera para representar la conexión rígida ideal con el terreno, y como nodo base para la aplicación de impedancias de suelo que simulan el acoplamiento con el terreno.

En el Método de Elementos Finitos (MEF), los elementos finitos se unen en un número finito de puntos llamados nodos. Los desplazamientos de los nudos son las incógnitas del problema. Los grados de libertad incluyen desplazamientos y, para elementos como vigas, también giros o rotaciones (Celigüeta-Lizarza, 2011). Las restricciones en los elementos pueden abarcar más de un grado de libertad y su uso resulta en una reducción en el número de GL a resolver (McGuire et al., 2000).

El MEF discretiza el dominio físico continuo en elementos de geometría simple (vigas, placas, sólidos) conectados en nodos, transformando el sistema de ecuaciones diferenciales parciales en un sistema algebraico de ecuaciones simultáneas que relaciona desplazamientos nodales con fuerzas aplicadas. Cada elemento relaciona los desplazamientos y los esfuerzos para después integrarse en una matriz global. En este caso, se emplea para modelar marcos en dos direcciones mediante elementos tipo barra 3D. Esta formulación incorpora las condiciones de frontera mediante la eliminación o modificación de grados de libertad restringidos, representa cargas sísmicas variables como vectores de fuerza dependientes del tiempo, y modela geometrías estructurales arbitrarias mediante mallas tridimensionales de elementos finitos (Celigüeta-Lizarza, 2011).

Para el análisis dinámico se consideran las matrices de masa, amortiguamiento y rigidez para formular

las ecuaciones de movimiento (Celigüeta-Lizarza, 2011). Uno de los fenómenos que aparece durante un movimiento sísmico es la Interacción Suelo–Estructura (ISE). La ISE describe el proceso mediante el cual la respuesta sísmica de una estructura se ve modificada por las propiedades dinámicas del suelo que la sostiene, y viceversa. Este enfoque concibe suelo y estructura como un sistema interrelacionado, en el que existe retroalimentación dinámica mutua. Su efecto se refleja en la variación de frecuencias, ajuste de los modos de vibración y cambios en desplazamientos y esfuerzos internos. Su importancia es elevada en suelos blandos, donde la amplificación y la extensión de la duración del movimiento sísmico incrementan la demanda sobre las estructuras.

En el diseño estructural en México, los espectros de diseño establecidos en las NTC-Sismo 2020 se construyen mediante un análisis probabilista de peligro sísmico, que integra la sismicidad regional, las leyes de atenuación del movimiento del terreno y los efectos de sitio locales. A partir de este análisis se determina la demanda sísmica asociada a una probabilidad de excedencia definida en un período de retorno determinado, expresada en términos del período predominante del sitio y el período fundamental de la estructura (Avilés y Pérez-Rocha, 1998). Estos espectros de respuesta constituyen la envolvente de demanda sísmica de diseño y son esenciales para el análisis y dimensionamiento de estructuras en la Ciudad de México.

Estos espectros se pueden expresar en función del desplazamiento, velocidad o aceleración en función del período, en términos del período predominante del sitio y el período fundamental de la estructura (Avilés y Pérez-Rocha, 1998; Avilés y Pérez-Rocha, 2010). Son esenciales para el análisis espectral y permiten estimar la demanda sísmica sobre estructuras con distintas propiedades geométricas y dinámicas. En esta tesis se emplean tanto los espectros normativos de la NTC-CDMX como espectros derivados de registros reales representativos de la Zona III.

La demanda sísmica depende directamente de las propiedades dinámicas del sistema estructural (Yigit Isbilibiroglu y Bielak, 2015), por lo que es necesario identificar las frecuencias naturales y modos de vibrar mediante un análisis modal. El análisis modal es un procedimiento matemático que calcula las frecuencias naturales y modos de vibración de la estructura mediante la resolución del problema de autovalores generalizados $[K - \omega^2 M]\varphi = 0$.

Este análisis es el punto de inicio para el análisis modal espectral y para la correcta interpretación de la respuesta dinámica durante un sismo (Wada, 1965; Yigit Isbilibroglu y Bielak, 2015), especialmente en la Zona Metropolitana del Valle de México (por su acrónimo: ZMVM) donde los posibles efectos de resonancia hacen necesaria la identificación precisa de los períodos fundamentales. El análisis modal espectral utiliza las frecuencias y modos de vibración obtenidos para calcular los desplazamientos espectrales máximos mediante la interpolación del espectro de diseño en cada período modal T_i , aplicando posteriormente la combinación SRSS para obtener la respuesta total de la estructura (Villafuerte Lujano y Fernández-Dávila Gonzales, 2024).

La metodología utilizada se basa en un modelo matemático que busca el equilibrio dinámico de una estructura. Con ese fin es necesario especificar las condiciones de frontera del modelo, ya que errores en su planteamiento podrían invalidar la respuesta obtenida. Estas condiciones son restricciones aplicadas en nodos y en los extremos de las barras que definen el comportamiento de éstas ante cargas externas. Algunas de las condiciones de apoyo más comunes son el apoyo simple, el apoyo móvil y el empotramiento (Celigüeta-Lizarza, 2011).

Además de las condiciones de frontera, el comportamiento dinámico de la estructura está influido por la forma en que las ondas sísmicas se propagan a través del suelo en el que está construida. En la ZMVM, y en particular en las zonas con suelos blandos de la CDMX, se presentan condiciones particulares debido a los sedimentos de los antiguos lagos (Grupo de Trabajo de la Red Sísmica de la Ciudad de México, 2021). Los efectos de suelo han producido amplificaciones en las arcillas de baja velocidad incluso con sismos ocurridos a más de 300 km de distancia (Quinde y Reinoso, 2020).

Algunas simulaciones numéricas han mostrado que la amplificación y generación de ondas superficiales en los bordes de la cuenca del lago han contribuido a la larga duración de los movimientos en la ZMVM (Cruz-Atienza et al., 2016), debida a la refracción y el atrapamiento de las ondas sísmicas dentro de la cuenca. Un aspecto adicional que modifica el comportamiento del suelo es la rigidez dinámica, que incluye la deformación y el amortiguamiento bajo cargas dinámicas.

La rigidez dinámica está relacionada con los efectos de sitio e influye directamente en cómo el suelo amplifica el movimiento sísmico. La amplificación se presenta cuando la onda sísmica se propaga de un medio

con velocidad de propagación rápida a uno con velocidad más baja. Los contrastes de impedancia y la conservación de la energía provocan que la amplitud de la onda aumente (Chávez-García, 2014).

En el modelo numérico desarrollado en MATLAB, la rigidez dinámica del suelo se representa mediante resortes traslacionales y rotacionales ubicados en los nodos de base, cuyos coeficientes de rigidez se calculan mediante las expresiones de Gazetas para cimentación superficial sobre semiespacio elástico. Esta formulación equivalente permite simular distintos tipos de terreno (Zonas II y III según NTC-Sismo 2017) sin requerir la discretización volumétrica completa del medio continuo.

La cimentación de una estructura representa la interfaz a través de la cual la energía sísmica es transmitida del suelo a la estructura, y en sentido contrario, a través de la cual la estructura puede disipar energía hacia el suelo (Chávez-García, 2014). La interfaz suelo-cimentación transmite energía en ambas direcciones: del suelo hacia la estructura durante la excitación sísmica (interacción cinemática), y de la estructura hacia el suelo mediante radiación de ondas generadas por las vibraciones estructurales (amortiguamiento por radiación geométrica). A diferencia de una conexión infinitamente rígida, la interfaz puede modelarse con condiciones que permitan reflejar el comportamiento real del suelo, donde el movimiento de la cimentación difiere del movimiento de campo libre como resultado de los efectos cinemáticos y de la flexibilidad del sistema (Pinto-Vega y Ledezma, 2019).

La interacción suelo-estructura se representa mediante elementos de rigidez y amortiguamiento equivalentes en la base de la estructura. Los coeficientes de rigidez (K_b , K_r) y amortiguamiento (C_b , C_r) se calculan usando las expresiones analíticas de Gazetas para cimentaciones superficiales sobre semiespacio elástico. Esta formulación equivalente captura los efectos dinámicos principales de la ISE: el alargamiento del período fundamental debido a la flexibilidad de la base, el incremento del amortiguamiento del sistema por radiación geométrica, y la redistribución de fuerzas internas en la superestructura. El método de superposición permite desacoplar el problema en dos subsistemas independientes, evitando la discretización volumétrica completa del medio continuo que requeriría modelos tridimensionales de elementos finitos con decenas de miles de grados de libertad (Avilés López et al., 2016).

El método de superposición desacopla el sistema acoplado suelo-estructura en dos subsistemas: el movimien-

to de campo libre y el movimiento inercial de la superestructura, cuya suma proporciona el desplazamiento total (Avilés López et al., 2016). Se expresa mediante la siguiente ecuación:

$$U(t) = U_1(t) + U_2(t) \quad (1)$$

donde:

- $U(t)$ = desplazamiento total del sistema acoplado
- $U_1(t)$ = desplazamiento de la base estructural bajo el movimiento del terreno (interacción cinemática)
- $U_2(t)$ = desplazamiento relativo de la superestructura respecto a la base (interacción inercial)

Al sustituir en la Ecuación de Movimiento:

$$M\ddot{U} + C\dot{U} + KU = -M\ddot{U}_1 \quad (2)$$

por lo tanto queda:

$$M\ddot{U}_2 + C\dot{U}_2 + KU_2 = -M\ddot{U}_1 \quad (3)$$

Esta formulación reproduce los efectos dinámicos de la flexibilidad del suelo (alargamiento de período, incremento de amortiguamiento) mediante rigideces equivalentes calculadas analíticamente, evitando la discretización volumétrica completa del medio continuo.

0.6. MARCO TEÓRICO

Las teorías que se han aplicado en los suelos blandos de la Zona Metropolitana del Valle de México están basadas en datos numéricos y en observaciones de campo. Este apartado presenta los antecedentes teóricos de la Interacción Suelo-Estructura (ISE) y los casos históricos que fundamentan la metodología adoptada en este trabajo.

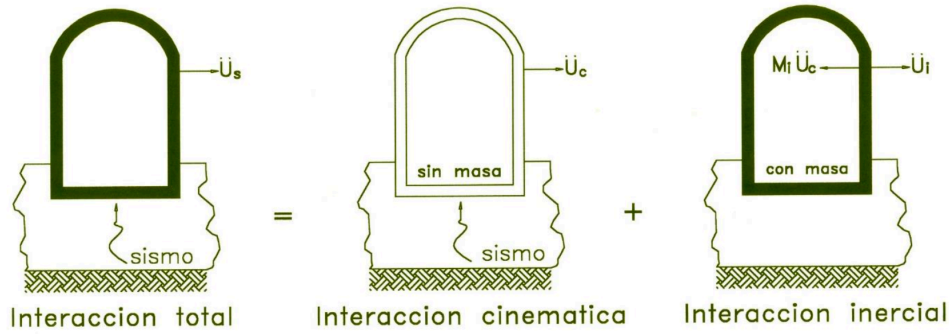


Figura 1: Esquema del fenómeno en interacción cinemática e inercial (Avilés López et al., 2016).

La complejidad de la Interacción Suelo-Estructura se basa en la interacción de dos componentes interdependientes que se deben modelar de forma diferenciada: la interacción cinemática y la interacción inercial. La interacción inercial se manifiesta principalmente cuando la masa de la estructura induce deformaciones significativas en el suelo deformable; esta variante es directamente dependiente de las diferencias de rigidez entre el suelo y la estructura (Avilés López et al., 2016). Esta interacción produce un alargamiento del período fundamental del sistema debido a la mayor flexibilidad del conjunto suelo-estructura respecto a la estructura aislada; a la vez, el amortiguamiento del sistema cambia, pudiendo incrementarse por efectos de radiación y disipación de energía en el suelo, llegando incluso a afectar la disipación de energía estructural si el amortiguamiento de la estructura pierde efectividad. Por otro lado, la interacción cinemática se produce por la incapacidad de la cimentación para replicar exactamente el movimiento sísmico en su vecindad inmediata. La superposición de las ondas sísmicas incidentes con las reflectadas y difractadas alrededor de la cimentación produce un movimiento de entrada a la estructura distinto al que ocurriría en campo libre. Este proceso puede reducir la amplitud de las traslaciones y producir efectos de torsión y cabeceo debido a la heterogeneidad del campo de movimiento (Avilés López et al., 2016).

El análisis de ISE se realiza en tres fases: primero se calcula el movimiento cinemático; después se estiman las funciones de impedancia; en el tercer paso se modela la respuesta estructural sobre un campo flexible. Si bien se suele considerar el efecto de la ISE sobre el período del sistema, su impacto en la ductilidad estructural también es relevante en suelos blandos, donde la deformabilidad del terreno puede modificar la demanda sísmica impuesta a la estructura, llegando a incrementar la cortante basal y alterar el estado límite de desempeño esperado. Las Normas Técnicas Mexicanas y el Manual de Diseño por Sismo de la CFE han

integrado criterios para suelos blandos, aunque pueden omitir la interacción cinemática detallada, lo que podría subestimar la respuesta real de las estructuras (Avilés López et al., 2016). En estructuras que operan en régimen elástico la ISE puede reducir la respuesta del sistema respecto a la de base fija; en estructuras inelásticas, en cambio, la interacción puede ampliar la demanda de ductilidad y los desplazamientos laterales. Las amplificaciones del movimiento basal son de mayor importancia cuando hay contrastes de rigidez entre estratos, pudiendo propiciar condiciones de resonancia que modifican la respuesta sísmica de la estructura, como se observó en el sismo de Kobe de 1995 (Vicencio et al., 2023). El sismo de Kobe produjo el alargamiento del período del puente Hanshin, donde la flexibilidad de la cimentación incrementó las fuerzas estructurales en el puente (Vicencio et al., 2023).

La resonancia entre el período del edificio y el período amplificado del terreno puede intensificarse por la ISE, llegando a comprometer la seguridad estructural. En la Ciudad de México, durante el sismo del 19 de septiembre de 1985, la coincidencia entre el período de las estructuras y el del terreno provocó resonancia que afectó numerosas edificaciones (Vicencio et al., 2023). Para modelar la transferencia energética entre el suelo y la estructura en sistemas más complejos, el análisis de Fourier y el método de la respuesta compleja en frecuencia son herramientas necesarias (Avilés López et al., 2016).

En el Valle de México, la resonancia y amplificación se deben a las características del subsuelo, compuesto por depósitos lacustres de hasta 70 metros de arcillas altamente compresibles con baja resistencia al corte y tiempos de consolidación prolongados (Quinde y Reinoso, 2020). Esto provoca grandes variaciones de rigidez y resistencia en distancias cortas e impide la generalización de parámetros promedio aplicables a toda la Ciudad, lo que subraya la necesidad de perfiles locales detallados para un modelado numérico preciso. La Zona III (Zona de Lago), definida por el Reglamento de Construcción de la Ciudad de México, se caracteriza por un suelo blando que amplifica las ondas sísmicas de períodos largos. Los perfiles de velocidad de corte (ondas S) en este sitio pueden llegar a valores inferiores a 30 m/s, colocando a este suelo en la categoría más severa de suelos blandos según la normativa de la American Society of Civil Engineers (ASCE) (Barnaure y Manoli, 2019; Khosravikia et al., 2018). Las amplificaciones absolutas en la Zona de Lago podrían alcanzar valores de 100 a 500 veces, entre las más altas reportadas a nivel mundial (Cruz-Atienza et al., 2016). El hundimiento regional por consolidación del suelo empeora adicionalmente las condiciones dinámicas, al

modificar los espesores efectivos y complejizar la interacción entre el suelo y la cimentación.

Con base en los fundamentos teóricos expuestos, se adopta un enfoque que integra el método de la matriz de rigidez dentro del marco del Método de Elementos Finitos para representar el acoplamiento suelo-estructura a través de rigideces equivalentes en los nodos de la base. De esta manera es posible capturar los efectos de interacción inercial y cinemática sin discretizar el medio continuo del suelo, lo que resulta adecuado para el análisis de edificaciones sobre suelos lacustres como los de la Zona III del Valle de México.

0.7. ALCANCE Y DELIMITACIONES DEL ESTUDIO

La metodología del acoplamiento se implementa mediante elementos de rigidez y amortiguamiento en los apoyos, lo que permite capturar la flexibilidad dinámica del suelo sin discretizar su dominio. No se incluyen condiciones no lineales ni otro tipo de disipaciones; el trabajo se centra exclusivamente en el acoplamiento elástico lineal. Para el análisis modal y el análisis modal espectral se utilizan como referencia principal los registros de los terremotos del 19 de septiembre de Michoacán (1985) y Puebla-Morelos (2017). Los resultados se contrastan con modelos equivalentes elaborados en SAP2000 para verificar la respuesta provista por el código en MATLAB.

Este trabajo tiene un enfoque computacional, sin ensayos de laboratorio ni instrumentación en campo. El análisis se restringe a edificios tipo marco en dos direcciones; se excluyen puentes, túneles, cimentaciones profundas y cualquier otra infraestructura de características más complejas. Para el modelo geotécnico se utilizan únicamente datos publicados; no se ejecutaron campañas nuevas de reconocimiento ni prueba *in situ* a la fecha de publicación del trabajo.

El análisis sísmico emplea exclusivamente los eventos sísmicos de septiembre de 1985 y 2017, elegidos por la extensa documentación disponible sobre ambos. La técnica de acoplamiento queda limitada a rigideces equivalentes en los apoyos empotrados. Este trabajo no plantea reformas reglamentarias formales ni modificaciones a las Normas Técnicas Complementarias vigentes a la fecha de publicación. Aunque se representa la influencia del suelo mediante elementos de frontera flexibles en los apoyos, no se modela de manera explícita el terreno con elementos de volumen, ni se amplía el estudio más allá de los alcances

numéricos mencionados.

1

Métodos de Análisis Estructural

I.1. CARACTERÍSTICAS DE LA SISMICIDAD EN REGIONES DE ALTA VULNERABILIDAD ESTRUCTURAL

El riesgo sísmico en zonas de alta vulnerabilidad estructural depende no solo de la magnitud de los eventos sino también de la geología local, las propiedades dinámicas del suelo y los fenómenos de acoplamiento entre ondas sísmicas y estructuras. La Ciudad de México es un caso de estudio destacado ya que, como se mencionó anteriormente, se asienta sobre antiguos depósitos de arcilla altamente compresibles. Esta condición del subsuelo produce una amplificación del movimiento sísmico y una prolongación de la duración del mismo.

La duración de los sismos en este tipo de suelos blandos se asocia también a mecanismos complejos como reflexiones internas, dispersión de ondas superficiales y efectos de borde. En cuencas sedimentarias los modos de vibrar del suelo tienden a dominar la respuesta sísmica, desplazando la energía a frecuencias más bajas, incrementando la interacción resonante con estructuras de mediana y gran altura (Avilés López et al., 2016).

El acoplamiento entre los modos de vibrar del suelo y el de las estructuras generan una transferencia de energía mucho más eficiente, llegando a duplicar la duración efectiva del movimiento en comparación con zonas de roca (Avilés López et al., 2016). Este fenómeno redistribuye la energía en componentes rotacionales y torsionales, difíciles de vislumbrar en los análisis a través del uso de espectros convencionales. La impedancia que se da debido a las rocas volcánicas subyacentes, que tienen una velocidad de ondas de corte de, al menos, 1300m/s genera la reflexión de las ondas sísmicas en la cuenca lacustre, intensificando la energía en la superficie y produciendo resonancias locales persistentes (Quinde y Reinoso, 2020).

Son estas condiciones las que propician una alteración de las frecuencias del movimiento sísmico. En la Zona III (Zona de Lago) de la Ciudad de México se han registrado niveles de amplificación espectral de entre 10 y 50 veces en frecuencias de 0.2 a 0.7 Hz, en comparación con las otras Zonas para eventos de subducción (Cruz-Atienza et al., 2016).

El fenómeno ha sido consistente y se ha corroborado a través de simulaciones numéricas 3D, que muestra cómo la geología del valle y la geometría del depósito sedimentario controla la propagación de las ondas, provocando movimientos que se prolongan por más de 3 minutos en la Zona de Lago (Cruz-Atienza et al., 2016). En los modelos numéricos tridimensionales se ha observado que la geometría de la cuenca sedimentaria puede atrapar energía sísmica y redirigirla hacia zonas específicas, las cuales muestran patrones de daño recurrente. Por ejemplo, los efectos de dispersión lateral implican que la duración del movimiento no es uniforme incluso dentro de la misma zona de suelo blando. Por ejemplo, en edificios separados por aproximadamente 200 metros es posible experimentar diferencias de duración de hasta 30 segundos. Estas observaciones entre las duraciones deben incorporarse en mapas de microzonificación con el fin de poder representar de una manera adecuada el movimiento sísmico (Avilés López et al., 2016).

Estas observaciones se hicieron particularmente después del sismo del 19 de Septiembre de 1985, donde, a

pesar de las aceleraciones máximas relativamente bajas ($< 0.25g$) comparados con otros movimientos, el daño estructural presentado fue alto, principalmente debido a la duración prolongada del movimiento y a la coincidencia entre el período del suelo y el de edificaciones de mediana altura. En ese evento sísmico, la estación SCT registró un período dominante cercano a los 2 segundos, o $0.5Hz$, consecuencia de un acoplamiento resonante con estructuras de 8 a 15 niveles. La mayor parte de demanda espectral se ubicó entre $0.3Hz$ y $1Hz$, período frecuente en rascacielos y edificios de concreto reforzado de mediana altura (Diego Melgar y Çelebi, 2018). Además de la amplificación por la coincidencia de frecuencias, el daño fue acentuado debido al alargamiento provocado por la interacción suelo-estructura. La frecuencia obtenida del movimiento registrado en la Zona de Lago provino de los modos superiores de Rayleigh generados dentro de la cuenca y no como tal, de la fuente sísmica del evento.

Estas observaciones fueron confirmadas durante el sismo del 19 de septiembre de 2017, donde el peligro de los suelos blandos no proviene de fallas bajo la ciudad, sino de la amplificación y la duración de las ondas. El factor determinante del daño observado fue el fenómeno de resonancia entre el período dominante del sitio ($T_s 2.0s$) y el período fundamental de estructuras de mediana altura (8-15 niveles), amplificado por efectos de interacción suelo-estructura que alargaron los períodos estructurales hasta en un 30 – 50 % (Diego Melgar y Çelebi, 2018).

La Zona II - Zona de Transición es sensible a los sismos profundos de falla normal, o aquellos generados por fallas cercanas. En la Ciudad de México, esta zona se caracteriza por depósitos de arcillas más rígidas en comparación con las de la Zona de Lago (Avilés y Pérez-Rocha, 2010; Efraín Ovando-Shelley y Santoyo, 2013). El fenómeno de resonancia sigue siendo relevante en esta zona, aunque con características distintas a las de la Zona de Lago. En zonas similares en la ciudad de Santiago de Chile, los gradientes de rigidez provocados por estratos de arcillas y gravas generan distribuciones asimétricas de la energía sísmica que intensifican el impacto en las edificaciones (Pinto-Vega y Ledezma, 2019).

1.2. INFLUENCIA DE LA SISMICIDAD EN EL DISEÑO ESTRUCTURAL

Los modelos estructurales convencionales parten de una configuración sin deformaciones iniciales. Esta suposición resulta inadecuada en edificaciones asentadas sobre depósitos lacustres con hundimientos diferenciales, donde la respuesta dinámica bajo eventos sísmicos se ve alterada por el estado de esfuerzos preexistente (Avilés y Pérez-Rocha, 2010).

Los esfuerzos internos generados por el fenómeno de subsidencia generan un estado de esfuerzos no considerado inicialmente en los modelos tradicionales. Estos esfuerzos generados cambian la forma en la que los edificios se comportan ante cargas sísmicas, cambiando los modos vibratorios y generando trayectorias de carga excéntricas, incluso hasta en estructuras regulares y geoméricamente simétricas (Avilés y Pérez-Rocha, 2010). Esta interacción ha provocado colapsos donde las estructuras no fallaron por mal diseño sino por la deformación diferencial acumulada a lo largo del tiempo. Se ha observado que la existencia de grietas y reconsolidación en el terreno altera la rigidez de la edificación, generando modos de vibrar no previstos.

Como menciona Efraín Ovando-Shelley y Santoyo, 2013, muchos edificios experimentan estos fenómenos incluso antes de que ocurran los sismos, resultado de los eventos pasados. Estos hechos observados en la Ciudad de México hacen que los modelos de análisis subestimen la realidad de la estructura, pues no logran integrar al proceso de modelado la flexibilidad real de la Interacción Suelo-Estructura. La duración sísmica observada en la ZMVM puede superar los 180 segundos, mientras que la mayoría de los modelos se calibran con registros de 30 segundos. Esta discrepancia implica que la demanda sísmica real no queda representada solo por las aceleraciones máximas del evento; la duración prolongada alarga la respuesta estructural y puede acercar el período efectivo de la estructura al período del terreno, incrementando la transferencia de energía y el riesgo de resonancia (Avilés López et al., 2016; Cruz-Atienza et al., 2016; Quinde y Reinoso, 2020). Es importante mencionar que “Las máximas amplificaciones se presentan cuando se tiene roca basal perfectamente rígida, debido a la ausencia de amortiguamiento geométrico” (Avilés López et al., 2016).

La caracterización de la demanda sísmica con espectros de respuesta tomados de estaciones ubicadas en zonas con suelo totalmente rígido han sido insuficientes para tener un entendimiento completo de los eventos sísmicos. Esto es debido a que las estaciones de roca no representan la duración ni la intensidad

del movimiento de las estructuras ubicadas en la Zona III. Durante el sismo del 19 de septiembre de 1985 los espectros de respuesta de la estación SCT, ubicada en el antiguo edificio de la Secretaría de Comunicaciones y Obras Públicas, construido en 1954, excedieron los valores normativos del Reglamento de Construcciones y de los Servicios Públicos en el Distrito Federal de 1952 (Leonardo Ramírez Guzmán, 2022; Ramos Jiménez, 2008).

No fue sino hasta 2004 cuando el Reglamento de Construcciones para el Distrito Federal incrementó hasta 2.5 veces la ordenada espectral para períodos de 1s a 3s ($1Hz$ y $\approx 0.33Hz$) en la Zona de Lago respecto a los suelos de roca. A pesar de estas medidas correctivas han mostrado validez en eventos posteriores, esto no quiere decir que los espectros de respuesta sean por sí solos suficientes (Diego Melgar y Çelebi, 2018). "Los espectros para zonas como la III de la ZMVM deben incluir factores correctivos por comportamiento no lineal del suelo, ductilidad y sobrerresistencia, en lugar de simples factores de amplificación genéricos (Avilés López et al., 2016).

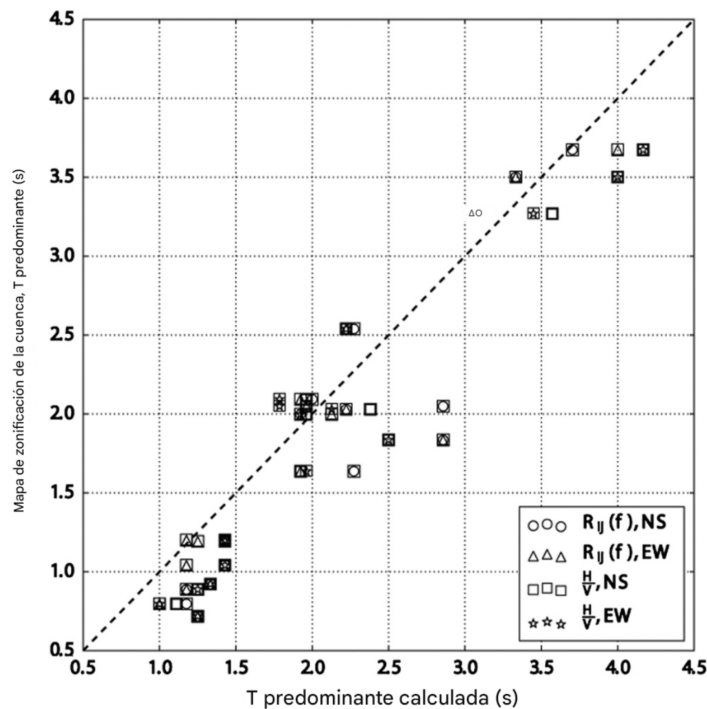


Figura 1.1: Comparación entre los periodos predominantes calculados mediante relaciones R_{ij} y cocientes espectrales H/V , frente a los valores asignados por el mapa de zonificación sísmica del RCCDMX. Adaptado de Diego Melgar y Çelebi, 2018

La zonificación sísmica utilizada en la ZMVM, definitivamente ha evolucionado al incorporar los detalles

geotécnicos del terreno. La figura 1.1 muestra la comparación entre los periodos predominantes calculados mediante relaciones R_{ij} y cocientes espectrales H/V , frente a los valores asignados por el mapa de zonificación sísmica del Reglamento de Construcciones de la Ciudad de México del 2004. La línea punteada representa la referencia 1:1. La alineación cercana de varios puntos a esta diagonal podría sugerir una correspondencia razonable entre los valores medidos y los asignados por el reglamento, lo cual respalda la validez de utilizar espectros elásticos basados en el periodo dominante local como criterio de diseño. (Adaptado de Diego Melgar y Çelebi, 2018).

Esta filosofía de diseño permite darle la importancia adecuada a la Interacción Suelo-Estructura. “Los procedimientos de diseño deben incluir retroalimentación geotécnica y estructural, especialmente en regiones con subsidencia. Ignorar la Interacción Suelo-Estructura incluso puede resultar en diseños que no se enfocan adecuadamente en el costo-beneficio o ser directamente inseguros”(Khosravikia et al., 2018).

De la misma manera, la omisión de este fenómeno puede conducir a errores significativos con efectos especialmente dañinos en la estimación de las aceleraciones de los niveles superiores de las estructuras. El acoplamiento dinámico entre el suelo y ciudad son condiciones de frontera fundamentales que deben ser modelados de manera explícita. La simulación numérica con el Método de los Elementos Finitos y el uso de registros sísmicos calibrados con las condiciones locales deben de convertirse en prácticas más comunes para estructuras de Tipo A, según el Reglamento de Construcciones de la CDMX. El diseño estructural en la Zona Metropolitana del Valle de México requiere modelos que integren las condiciones geotécnicas locales y la interacción dinámica entre edificaciones y suelo.

1.3. MÉTODOS DE ANÁLISIS ESTRUCTURAL EN ZONAS SÍSMICAS

El diseño sismorresistente convencional asume que cada edificio actúa de manera aislada frente al movimiento del terreno. Esta suposición conduce a subestimar la interacción dinámica entre estructuras vecinas. En entornos urbanos con alta densidad de edificaciones, el fenómeno de Interacción Suelo-Estructura puede amplificar la demanda sísmica en función de la geometría, la rigidez y las condiciones del conjunto. No tomar en consideración este acoplamiento puede llevar a errores críticos. Prueba de esto son los registros de colapsos

provocados por sincronizaciones entre edificios que al ser evaluados de manera individual cumplían con la normativa vigente, sin embargo, terminaron colapsando tras presentar resonancia y amplificación durante el sismo (Vicencio et al., 2023).

A pesar de los avances que se han desarrollado en el ámbito del análisis y diseño estructural las herramientas disponibles y el uso normalizado ó normado en los proyectos no ha sido ampliamente difundido. Sin embargo, el análisis en el dominio del tiempo ha sido la herramienta principal al momento de determinar la interacción entre el suelo y los edificios con propiedades dinámicas distintas. A la par, el método de superposición modal ha visto una disminución en su uso ya que este pierde validez precisamente cuando el suelo introduce acoplamientos entre distintos sistemas. Se requieren modelos capaces de integrar el terreno y varias superestructuras en una misma formulación computacional.

Al día de hoy comienza a tomar fuerza una tendencia para emplear modelos híbridos de elementos finitos para responder a la necesidad de precisión y eficiencia en los análisis. Aunque los modelos de mucha precisión tienen un costo computacional alto, permiten capturar la física esencial de la Interacción Estructura-Suelo-Estructura (IESE y por sus siglas en inglés: SSSI) cuando se alimentan con datos de campo y se utilizan en conjunto con técnicas de reducción modal (Vicencio et al., 2023).

Para tratar estos sistemas con numerosos grados de libertad sin perder precisión, se requiere una formulación matricial eficiente. La base matemática que permitió resolver sistemas con numerosos grados de libertad fue el método de la Matriz de Rigidez (Gloyna, 1964). Este método matemático satisface los criterios de formulación del comportamiento dinámico y modal (Wada, 1965). Este método se fundamenta en la formulación energética de la deformación, de la cual se deriva una matriz de rigidez que relaciona los desplazamientos nodales con las fuerzas internas mediante la ley de Hooke generalizada. Para cada elemento estructural (viga o columna) se genera una matriz cuadrada simétrica de dimensión $n \times n$, donde n es el número de grados de libertad del elemento: 4 para elementos 2D (2 nodos \times 2 GL/nodo) y 12 para elementos 3D (2 nodos \times 6 GL/nodo) (Celigiüeta-Lizarza, 2011; Gloyna, 1964).

La formulación matricial concentra las masas en los nodos de cada entrepiso, asignando a cada uno tres grados de libertad traslacionales ($U_x, U_y, U_z, \theta_x, \theta_y, \theta_z$) que capturan el movimiento inercial del piso como

cuerpo rígido mediante el supuesto de diafragma rígido (Avilés López et al., 2016).

A su vez, se posibilita el cálculo de autovalores (*eigenvalues*) y autovectores (*eigenvectors*), permitiendo obtener las frecuencias naturales y modos de vibrar del sistema. La ortogonalidad de los modos de vibración respecto a las matrices de masa y rigidez desacopla las ecuaciones de movimiento, reduciendo el sistema de n ecuaciones acopladas a n ecuaciones independientes que se resuelven individualmente y se superponen para obtener la respuesta total. La forma matricial de las ecuaciones de movimiento:

$$M\ddot{u} + C\dot{u} + Ku = F(t) \quad (1.1)$$

es una evidencia de la influencia transmitida del Método de la Matriz de Rigidez al Método de los Elementos Finitos (MEF). Este enfoque es de suma importancia al analizar el comportamiento de una estructura sometida a fuerzas con variación en el tiempo.

A continuación, se explica como cada término de la ecuación es ensamblado:

- Matriz de masas $[M]$ Para un elemento finito, la matriz de masa se define como:

$$M^e = \int_v \rho N^T N dv \quad (1.2)$$

donde ρ es la densidad del material y N es la matriz de funciones de interpolación que describe la variación del campo de aceleraciones dentro del elemento. Generalmente, esta matriz no es diagonal, lo que indica un acoplamiento inercial entre grados de libertad. La matriz global de masas $[M]$ se obtiene ensamblando las matrices individuales M^e correspondientes a cada elemento (Celigiüeta-Lizarza, 2011).

- Matriz de amortiguamiento $[C]$

En esta matriz se modela la capacidad de disipación de energía en el sistema estructural. La matriz de amortiguamiento C^e puede ser proporcional a la matriz de masas del elemento ($C^e = \alpha M^e$) o proporcional a la matriz de rigidez del elemento ($C^e = \beta K^e$). Es una combinación lineal de las matrices de masa $[M]$ y de la matriz de rigidez $[K]$. El modelo combinado, conocido como amortiguamiento

proporcional o de Rayleigh y se expresa como:

$$C^e = \alpha M^e + \beta K^e \quad (1.3)$$

Al igual que la matriz de masas, la matriz de amortiguamiento global $[C]$ se obtiene ensamblando las matrices de amortiguamiento correspondientes a todos los elementos (Celigüeta-Lizarza, 2011).

- Matriz de Rigidez $[K]$

Modela capacidad que posee la estructura para oponerse al movimiento. La matriz de rigidez del elemento finito es una matriz cuadrada simétrica cuyo tamaño es igual al número de grados de libertad del elemento y es formulada a partir de la energía interna de deformación (Celigüeta-Lizarza, 2011). Además matriz de rigidez completa también se obtiene al ensamblar las matrices de los distintos elementos que componen a la estructura según los grados de libertad que corresponden a cada uno.

- Vectores de Desplazamiento: (\ddot{u}, \dot{u}, u) y la fuerza, representada como $F(t)$ o $P(t)$

Donde u es el vector de desplazamientos, \dot{u} el vector de velocidades, y \ddot{u} el vector de aceleraciones del sistema (Avilés López et al., 2016). $F(t)$ o $P(t)$ es el vector de fuerzas externas que varían arbitrariamente con el tiempo (Avilés López et al., 2016). Las fuerzas pueden incluir cargas aplicadas directamente en los nodos, fuerzas en la superficie o fuerzas debidas a deformaciones o tensiones iniciales (Celigüeta-Lizarza, 2011). Estos vectores se obtienen por el ensamble de las contribuciones de los distintos elementos que componen la estructura.

El proceso de formulación del Método de Elemento Finito (MEF) debe garantizar que: 1) Se satisfaga la compatibilidad cinemática, y 2) Se satisfaga el equilibrio nodal. Al ensamblar las ecuaciones de equilibrio de los elementos para formar las ecuaciones de la estructura, se impone el equilibrio de las fronteras de unión entre los elementos a la estructura. Este ensamble lleva a la formulación de la ecuación de movimiento (Avilés López et al., 2016):

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = F(t) \quad (1.4)$$

En el análisis modal se estudia la respuesta libre de la estructura. Esto se obtiene al eliminar el amortiguamiento y las cargas externas, reduciendo la ecuación de movimiento a:

$$M\ddot{u} + Ku = 0 \quad (1.5)$$

De esta manera, la solución de la ecuación es

$$(K - \omega^2 M) \varphi = 0 \quad (1.6)$$

y proporciona las frecuencias naturales y los modos de vibración de la estructura.

Una de las ventajas del análisis modal es la posibilidad de descomponer un sistema con varios grados de libertad en ecuaciones desacopladas, las cuales se pueden resolver como un oscilador de segundo orden (McGuire et al., 2000).

1.4. ESPECTROS DE DISEÑO SÍSMICO

1.4.1. ESPECTROS NORMATIVOS NTC-CDMX 2017

Para el análisis espectral se emplean los espectros de diseño elástico establecidos en las Normas Técnicas Complementarias para Diseño por Sismo (NTC-Diseño por Sismo)(Gobierno de la Ciudad de México, 2017), correspondientes a suelo Tipo II y Tipo III.

ZONA II (TRANSICIÓN) - ESPECTRO ELÁSTICO

- Aceleración máxima del terreno: $a_0 = 0.16 g$.
- Coeficiente sísmico: $c = 0.40$.
- Período característico del sitio: $T_s = 1.5 s$.

- Período inferior del espectro constante:

$$T_a = 0.2 T_s = 0.3 \text{ s} \quad (1.7)$$

- Período superior del espectro constante:

$$T_b = T_s = 1.5 \text{ s} \quad (1.8)$$

ZONA III (LAGO) ESPECTRO ELÁSTICO

- Aceleración máxima del terreno: $a_0 = 0.10 g$.
- Coeficiente sísmico: $c = 0.60$.
- Período característico del sitio: $T_s = 2.5 \text{ s}$.
- Período inferior del espectro constante:

$$T_a = 0.2 T_s = 0.5 \text{ s} \quad (1.9)$$

- Período superior del espectro constante:

$$T_b = T_s = 2.5 \text{ s} \quad (1.10)$$

La ordenada espectral de aceleración $S_a(T)$ se calcula conforme a las expresiones definidas en la normativa:

$$S_a(T) = \begin{cases} a_0 + (c - a_0) \frac{T}{T_a}, & \text{para } T < T_a \\ c, & \text{para } T_a \leq T \leq T_b \\ c \frac{T_b}{T}, & \text{para } T > T_b \end{cases} \quad (1.11)$$

1.5. ANÁLISIS ESTRUCTURAL BASADO EN LA MATRIZ DE RIGIDECES

El análisis estructural de sistemas continuos, por ejemplo como losas de cimentación es un procedimiento complejo. Debido a que las soluciones geométricas estaban muy limitadas por condiciones del contorno, el Método de los Elementos Finitos surgió como una herramienta fundamental (Celigüeta-Lizarza, 2011).

El primer paso a realizar, sin importar el tipo de análisis, es idealizar la estructura; con el fin de simplificar los elementos estructurales y definir las propiedades del material con el que están hechos. Entre estas propiedades a definir se encuentra el módulo de Young (E) y la relación de Poisson (ν), las propiedades de la sección transversal y el momento de inercia (I). El Método de Rigideces es especialmente útil para el análisis de estructuras complejas. El Método de Elementos finitos para análisis estructural establece que:

1. División del medio en Elementos Finitos

El medio, que tiene, en un principio, infinitas formas posibles de deformarse, se divide mediante pequeñas superficies en regiones continuas entre sí. Estos son los elementos finitos. Usualmente, los elementos utilizados tienen formas geométricas y normalizadas (Celigüeta-Lizarza, 2011).

2. Unión en los nodos

Los elementos finitos se conectan a través de un número finito de nodos o puntos específicos.

3. Desplazamientos como incógnitas básicas

Los desplazamientos que se presentan en estos nodos, serán las incógnitas básicas del problema. Cada desplazamiento del nodo es independiente y dicha reacción determina la configuración deformada de la estructura.

4. Funciones de forma N_i o de interpolación

U será el campo de desplazamientos. Será, también, una función vectorial que indica las deformaciones en cualquier punto del elemento. Su aproximación por elemento se mantiene de forma independiente. La función permite calcular el valor de cualquier desplazamiento interior de un punto del elemento a través de la interpolación de los desplazamientos de los nodos (Celigüeta-Lizarza, 2011).

Usualmente son polinomios y deben cumplir con la condición de tener un valor de 1 en el nodo 1 al que se están asociando, así como 0 en los nodos restantes del elemento.

5. Relación Desplazamiento-Nodal (Desplazamientos nodales del elemento)

El campo de las deformaciones u en el interior del elemento es aproximado a través del uso de los desplazamientos nodales de cada elemento (δ_e) y las funciones de interpolación (N_i).

Sin embargo, para la aplicación de esta hipótesis de manera adecuada, se deberán de cumplir con condiciones de compatibilidad en las fronteras entre los elementos. En el Método de Elementos Finitos implica que las funciones de interpolación puedan proporcionar la continuidad de los desplazamientos. En el caso de que sea un problema de orden superior, las primeras derivadas deberán de ser continuas. El Método de los Elementos Finitos permite modelar una amplia variedad de elementos estructurales, cada uno con una formulación específica de su matriz de rigidez.

A través de la condensación de la matriz de rigidez global es posible disminuir el número de grados de libertad en un modelo de elementos finitos, lo que disminuye el tiempo de cálculo. Esto permite reducir la velocidad de cálculo computacional en estructuras con muchos marcos, y, específicamente en el análisis de marcos en dos direcciones. El ensamblaje de la matriz de rigidez global es el proceso en el cual las matrices de rigidez locales elementos individuales $[K_l]$ se unen para obtener la matriz de rigidez global $[K_G]$ de toda la estructura. El proceso se realiza con el fin de obtener el sistema de ecuaciones completo, lo que permite relacionar las fuerzas externas aplicadas en la estructura. A continuación se describe el proceso para poder ensamblar la matriz (basado en la información presentada por Celigüeta-Lizarza, 2011):

1. Definición de los grados de libertad.
2. Definición de grados de libertad según su localización sus condiciones de frontera.
3. Generación de la matriz de rigidez según el elemento local.
4. Matriz de transformación de coordenadas de eje local a eje global (según sea necesario).
5. Condensación de la matriz de rigidez global.

EJEMPLO 1.A: MODELO ESTÁTICO

Determinar el desplazamiento del nodo 2 que se muestra en la figura 1.1 si se le aplica una carga horizontal hacia la izquierda de 10kN . Se utilizará el marco mostrado en la figura para ilustrar el proceso de ensamblaje de la matriz de rigidez global $[K]$ y la condensación estática, reducir el sistema de ecuaciones. Considerar $AE = 8 \cdot 10^3 [\text{kN} \cdot \text{m}]$ ((Hibbeler, 2022) adaptado con ejemplo propio).

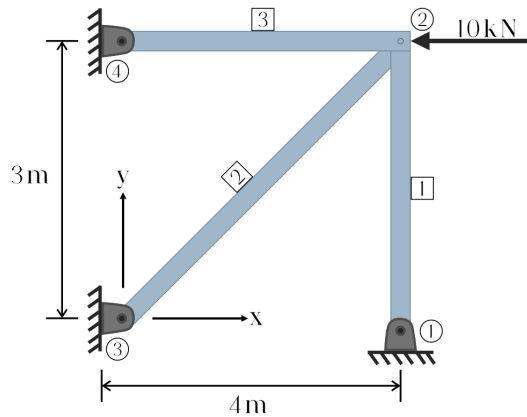


Figura 1.2: Ejemplo 1.A

- Identificación de nodos y grados de libertad

En un modelo de elementos finitos cada nodo posee grados de libertad asociados a los desplazamientos en las direcciones x y y , indicados como u_x y u_y , respectivamente. En este ejemplo, no se consideran rotaciones ni momentos, por lo que cada nodo tiene únicamente 2 grados de libertad: desplazamientos en x y en y .

Entonces, el vector de desplazamientos para este ejemplo es:

$$\delta = \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \\ u_{x4} \\ u_{y4} \end{bmatrix} \quad (I.12)$$

Se consideran únicamente desplazamientos traslacionales en u_x, u_y en cada nodo; para este ejemplo no se incluyen rotaciones.

- Matriz de rigidez local en coordenadas globales

En este tipo de barras, B se construye a partir de los cosenos directores del eje de la misma. De esta manera, la matriz de rigidez en coordenadas globales se construye a partir de la siguiente ecuación:

$$[K_L] = \frac{AE}{L} \begin{bmatrix} \cos^2(\theta) & \cos(\theta) \cdot \text{sen}(\theta) & -\cos^2(\theta) & -\cos(\theta) \cdot \text{sen}(\theta) \\ \cos(\theta) \cdot \text{sen}(\theta) & \text{sen}^2(\theta) & -\cos(\theta) \cdot \text{sen}(\theta) & -\text{sen}^2(\theta) \\ -\cos^2(\theta) & -\cos(\theta) \cdot \text{sen}(\theta) & \cos^2(\theta) & \cos(\theta) \cdot \text{sen}(\theta) \\ -\cos(\theta) \cdot \text{sen}(\theta) & -\text{sen}^2(\theta) & \cos(\theta) \cdot \text{sen}(\theta) & \text{sen}^2(\theta) \end{bmatrix} \quad (I.13)$$

donde:

- A = área de la sección transversal
- E = módulo de elasticidad
- L = longitud del elemento

- $\cos(\theta)$ y $\sen(\theta)$ de la barra

Para cada barra:

- Barra 1 (nodo 1 – nodo 2)

$L = 3m$, $\cos(\theta) = 0$, $\sen(\theta) = 1$, al sustituir en la Ecuación (1.13):

$$K_{L1} = \left(\frac{AE}{L}\right) \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (1.14)$$

$$= \frac{8 \cdot 10^3}{3} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (1.15)$$

$$K_{L1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2666.7 & 0 & -2666.7 \\ 0 & 0 & 0 & 0 \\ 0 & -2666.7 & 0 & 2666.7 \end{bmatrix} \quad (1.16)$$

- Barra 2 (nodo 2 – nodo 3)

$L = \sqrt{4^2 + 3^2} = 5m$, $\cos(\theta) = \frac{4}{5}$, $\sen(\theta) = \frac{3}{5}$, al sustituir en la Ecuación (1.13):

$$K_{L2} = \left(\frac{AE}{L}\right) \cdot \begin{bmatrix} \frac{16}{25} & \frac{12}{25} & -\frac{16}{25} & -\frac{12}{25} \\ \frac{12}{25} & \frac{9}{25} & -\frac{12}{25} & -\frac{9}{25} \\ -\frac{16}{25} & -\frac{12}{25} & \frac{16}{25} & \frac{12}{25} \\ -\frac{12}{25} & -\frac{9}{25} & \frac{12}{25} & \frac{9}{25} \end{bmatrix} \quad (1.17)$$

$$= \left(\frac{8 \cdot 10^3}{5} \right) \cdot \begin{bmatrix} 0.64 & 0.48 & -0.64 & -0.48 \\ 0.48 & 0.36 & -0.48 & -0.36 \\ -0.64 & -0.48 & 0.64 & 0.48 \\ -0.48 & -0.36 & 0.48 & 0.36 \end{bmatrix} \quad (1.18)$$

$$K_{L2} = \begin{bmatrix} 1024 & 768 & -1024 & -768 \\ 768 & 576 & -768 & -576 \\ -1024 & -768 & 1024 & 768 \\ -768 & -576 & 768 & 576 \end{bmatrix} \quad (1.19)$$

- Barra 3 (nodo 2 – nodo 4)

$L = 4m$, $\cos(\theta) = 1$, $\text{sen}(\theta) = 0$, al sustituir en la Ecuación (1.13):

$$K_{L3} = \left(\frac{AE}{L} \right) \cdot \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \left(\frac{8 \cdot 10^3}{4} \right) \cdot \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.20)$$

$$K_{L3} = \begin{bmatrix} 2000 & 0 & -2000 & 0 \\ 0 & 0 & 0 & 0 \\ -2000 & 0 & 2000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.21)$$

- Ensamblaje de la matriz global

Cada submatriz K_{Li} de tamaño 4×4 se inserta en la matriz global K_G , cuyo tamaño total es 8×8 debido a que el modelo cuenta con 4 nodos y 2 grados de libertad por nodo (u_x, u_y), esto resulta en un total de 8 grados de libertad en el sistema. Los grados de libertad involucrados para cada barra son:

- Barra 1 (nodo 1–2): DOF globales [1,2,3,4]
- Barra 2 (nodo 2–3): DOF globales [3,4,5,6]

- Barra 3 (nodo 2-4): DOF globales [3,4,7,8]

La matriz de rigidez global se obtiene sumando las contribuciones de cada barra. Se inicia estableciendo una matriz 8×8 de ceros y superponiendo en ella las matrices locales de cada barra, según los grados de libertad globales correspondientes.

Al sustituir en el vector de desplazamientos (Ec. (1.12)), previamente definido, los grados de libertad globales para cada barra son:

$$\text{Barra 1 (1-2)} \rightarrow [1,2,3,4], \quad \text{Barra 2 (2-3)} \rightarrow [3,4,5,6], \quad \text{Barra 3 (2-4)} \rightarrow [3,4,7,8]. \quad (1.22)$$

La contribución de cada barra a la matriz global puede escribirse como una matriz 8×8 con ceros en todas las entradas, salvo en las filas y columnas asociadas a sus grados de libertad. Por ejemplo:

$$\mathbf{K}^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2666.7 & 0 & -2666.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2666.7 & 0 & 2666.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (1.23)$$

$$\mathbf{K}^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1024 & 768 & -1024 & -768 & 0 & 0 \\ 0 & 0 & 768 & 576 & -768 & -576 & 0 & 0 \\ 0 & 0 & -1024 & -768 & 1024 & 768 & 0 & 0 \\ 0 & 0 & -768 & -576 & 768 & 576 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (1.24)$$

$$\mathbf{K}^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2000 & 0 & 0 & 0 & -2000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2000 & 0 & 0 & 0 & 2000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.25)$$

La matriz de rigidez global es entonces la suma de las tres contribuciones de las matrices locales:

$$\mathbf{K}_G = \mathbf{K}^{(1)} + \mathbf{K}^{(2)} + \mathbf{K}^{(3)}. \quad (1.26)$$

Al añadir cada matriz local en \mathbf{K}_G se obtiene:

$$\mathbf{K}_G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2666.7 & 0 & -2666.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3024 & 768 & -1024 & -768 & -2000 & 0 \\ 0 & -2666.7 & 768 & 3242.7 & -768 & -576 & 0 & 0 \\ 0 & 0 & -1024 & -768 & 1024 & 768 & 0 & 0 \\ 0 & 0 & -768 & -576 & 768 & 576 & 0 & 0 \\ 0 & 0 & -2000 & 0 & 0 & 0 & 2000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1.27)$$

- Condensación estática

Los nodos 1, 3 y 4 se encuentran completamente restringidos, por lo que los únicos grados de libertad desconocidos son u_{x2} y u_{y2} , correspondientes a las filas y columnas 3 y 4 de K_G . La matriz reducida queda entonces:

$$\mathbf{K}_{GR} = \begin{bmatrix} K_{33} & K_{34} \\ K_{43} & K_{44} \end{bmatrix} = \begin{bmatrix} 3024 & 768 \\ 768 & 3243 \end{bmatrix}. \quad (1.28)$$

La carga horizontal aplicada en el nodo 2 es de $10kN$ hacia la izquierda, por lo que el vector de fuerzas es:

$$\mathbf{F}_R = \begin{bmatrix} -10 \\ 0 \end{bmatrix} \text{ kN}. \quad (1.29)$$

El sistema reducido de ecuaciones se expresa como:

$$\mathbf{K}_{GR} \cdot \delta_R = \mathbf{F}_R, \quad \delta_R = \begin{bmatrix} u_{x2} \\ u_{y2} \end{bmatrix}.$$

El sistema reducido es

$$\begin{bmatrix} 3024 & 768 \\ 768 & 3243 \end{bmatrix} \begin{bmatrix} u_{x2} \\ u_{y2} \end{bmatrix} = \begin{bmatrix} -10 \\ 0 \end{bmatrix}. \quad (1.30)$$

De manera que al despejar, la matriz de rigidez reducida es

$$\mathbf{K}_{GR} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} = \begin{bmatrix} 3024 & 768 \\ 768 & 3243 \end{bmatrix}, \quad (1.31)$$

con un determinante de

$$\det(\mathbf{K}_{GR}) = k_{11}k_{22} - k_{12}k_{21} = 3024 \cdot 3243 - 768 \cdot 768 \quad (1.32)$$

$$\det(\mathbf{K}_{GR}) = 9\,217\,008. \quad (1.33)$$

La inversa de una matriz 2×2 es

$$\mathbf{K}_{GR}^{-1} = \frac{1}{\det(\mathbf{K}_{GR})} \begin{bmatrix} k_{22} & -k_{12} \\ -k_{21} & k_{11} \end{bmatrix}, \quad (1.34)$$

por lo que al sustituir los valores numéricos:

$$\mathbf{K}_{GR}^{-1} = \frac{1}{9\,217\,008} \begin{bmatrix} 3243 & -768 \\ -768 & 3024 \end{bmatrix}. \quad (1.35)$$

El vector de desplazamientos se obtiene como

$$\delta_R = \begin{bmatrix} u_{x2} \\ u_{y2} \end{bmatrix} = \mathbf{K}_{GR}^{-1} \mathbf{F}_R = \frac{1}{9\,217\,008} \begin{bmatrix} 3243 & -768 \\ -768 & 3024 \end{bmatrix} \begin{bmatrix} -10 \\ 0 \end{bmatrix} \quad (1.36)$$

Realizando el producto matricial:

$$u_{x2} = \frac{1}{9\,217\,008} (3243 \cdot (-10) + (-768) \cdot 0) = \frac{-32\,430}{9\,217\,008} \approx -3.52 \times 10^{-3} \text{ m}, \quad (1.37)$$

$$u_{x2} = -3.52 \times 10^{-3} \text{ m}$$

$$u_{y2} = \frac{1}{9\,217\,008} ((-768) \cdot (-10) + 3024 \cdot 0) = \frac{7\,680}{9\,217\,008} \approx 0.83 \times 10^{-3} \text{ m}. \quad (1.38)$$

$$u_{y2} = 0.83 \times 10^{-3} \text{ m}.$$

Convirtiendo a milímetros:

$$u_{x2} = -3.52 \text{ mm}$$

$$u_{y2} = 0.83 \text{ mm}$$

Por lo tanto, el desplazamiento en el nodo 2 es de 3.5 mm hacia la izquierda (eje x) y 0.8 mm hacia arriba (eje y).

2

Implementación del Código para el Análisis Sísmico

2.1. DESCRIPCIÓN DE LAS FUNCIONES DEL CÓDIGO EN MATLAB

La implementación numérica del análisis sísmico se estructura en cinco scripts principales de MATLAB, cada uno con funciones específicas dentro del flujo de cálculo. Los scripts se ejecutan de forma secuencial: cada uno utiliza las variables generadas en el proceso anterior, lo que garantiza la consistencia del modelo y reduce los tiempos de cómputo.

La primera etapa consiste en el preprocesamiento geométrico, donde se definen los nodos, elementos y

propiedades mecánicas de la estructura. En la segunda etapa se ensamblan las matrices globales de rigidez y masa mediante transformación de coordenadas locales a globales y superposición de contribuciones elementales. La tercera etapa aplica las condiciones de frontera (empotramientos en la base) y reduce el sistema matricial eliminando los grados de libertad restringidos. La cuarta etapa resuelve los problemas estático, modal y espectral con el sistema reducido. Finalmente, la quinta etapa incorpora la flexibilidad del suelo mediante rigideces equivalentes calculadas con las expresiones de Gazetas, lo que permite evaluar los efectos de interacción suelo-estructura sin discretizar el medio continuo. En las Figuras 2.1 y 2.2 se muestra el diagrama general del código, donde se observa la estructura modular completa y la secuencia de ejecución de los scripts.

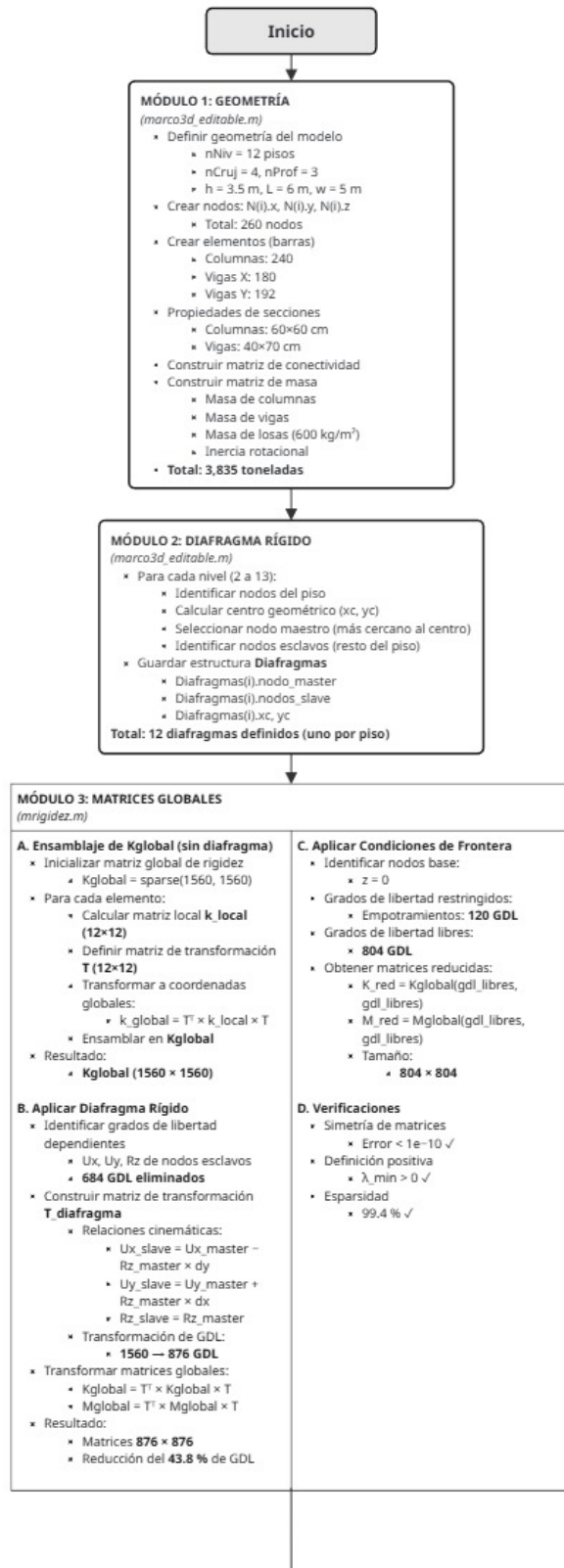
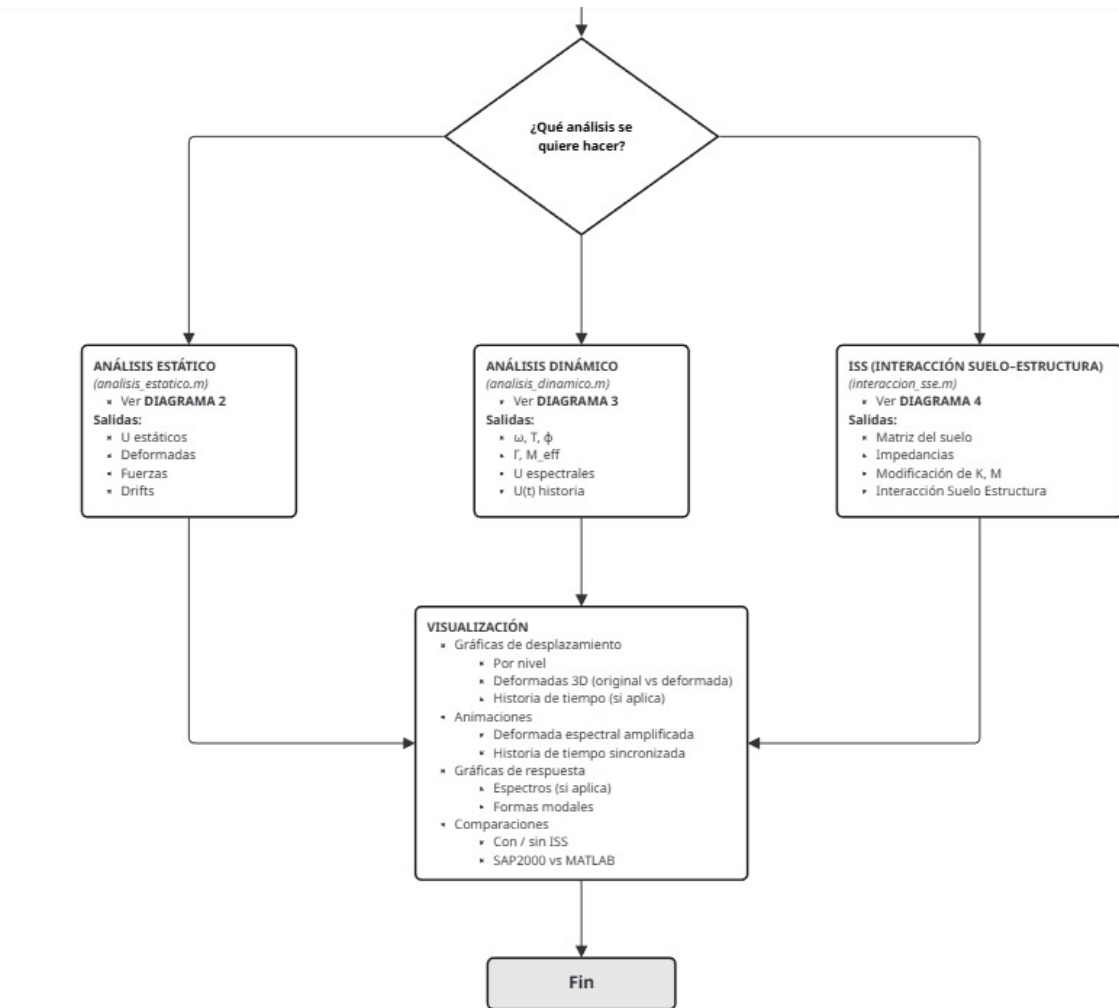


Figura 2.1: Diagrama General de decisiones de los scripts del código. Continúa en la siguiente página.



- NOTAS IMPORTANTES**
- FLUJO DE GDL**
 - 1560 (original) → 876 (con diafragma) → 804 (sin empotramientos)
 - VARIABLES CLAVE**
 - *gdl_mapping.T*: Matriz de transformación del diafragma
 - *gdl_libres*: Índices de GDL sin empotrar
 - *gdl_restringidos*: Índices de GDL empotrados
 - TRANSFORMACIONES**
 - Fuerzas:
 - $F_{\text{completo}} \rightarrow T^T \rightarrow F_{\text{reducido}}$
 - Desplazamientos:
 - $U_{\text{reducido}} \rightarrow T \rightarrow U_{\text{completo}}$
 - ARCHIVOS**
 - *marco3d_editable.m*:
 - Geometría + Masa + Diafragmas
 - *mrigidez.m*:
 - K global + Transformación + Condiciones de frontera
 - *analisis_estatico.m*:
 - Cargas + Resolución + Visualización
 - *analisis_dinamico.m*:
 - Modal + Espectral + Tiempo-historia

Figura 2.2: Continuación del Diagrama General de los scripts del código.

2.1.1. GEOMETRÍA (`marco3d_editable.m`)

Este script define la geometría tridimensional del marco estructural mediante la generación automática de nodos en una retícula ortogonal de tres dimensiones. Los parámetros básicos de la geometría del edificio son:

- Número de niveles
- Número de crujías en X
- Número de crujías en Y
- Longitud de los claros
- Altura del entrepiso

Con estos parámetros definidos, los nodos se generan mediante ciclos anidados que recorren cada nivel, crujía y profundidad del edificio.

La creación de nodos sigue un esquema de numeración sistemático que permite identificar de manera única cada posición en el espacio. Las coordenadas (x,y,z) de cada nodo se almacenan en una estructura de MATLAB que relaciona el número de nodo con su ubicación espacial. Esta indexación permite construir las matrices de conectividad y localizar los nodos de la base para la aplicación de condiciones de frontera.

Todas las propiedades mecánicas y geométricas de los elementos estructurales se definen considerando secciones regulares de concreto reforzado. La geometría de columnas y vigas es paramétrica y puede modificarse a conveniencia. A partir de estas dimensiones se calculan las áreas transversales, los momentos de inercia I_y e I_z , y la constante de torsión J mediante formulaciones para la geometría de las secciones. El módulo de elasticidad del concreto se establece en $E = 2.1 \times 10^{10}$ Pa, y el módulo de cortante se obtiene mediante la relación $G = E/[2(1 + \nu)]$, asumiendo un coeficiente de Poisson de $\nu = 0.3$.

La distribución del peso propio de la estructura se realiza mediante un proceso de tributación que genera masas concentradas en cada nodo. Las masas de las columnas se distribuyen equitativamente entre sus nodos extremos, mientras que las masas de las vigas y losas se asignan considerando el área tributaria de cada nodo.

Para el sistema de entrepiso se adopta una losa maciza de concreto de 25 cm de espesor, valor que genera una carga superficial de 600 kg/m^2 , calculada considerando un peso volumétrico del concreto de $2,400 \text{ kg/m}^3$. En el modelo, esta carga corresponde exclusivamente al peso propio de la losa, sin incluir acabados ni sobrecarga de uso, dado que el objetivo del análisis es la evaluación del comportamiento dinámico de la estructura bajo su peso propio. El resultado es una matriz de masas nodales (Figura 2.3), que contiene las toneladas distribuidas entre el total de nodos generados (Ghavami, 2015; McGuire et al., 2000). Los componentes de cada nodo se activan únicamente según los grados de libertad traslacionales.

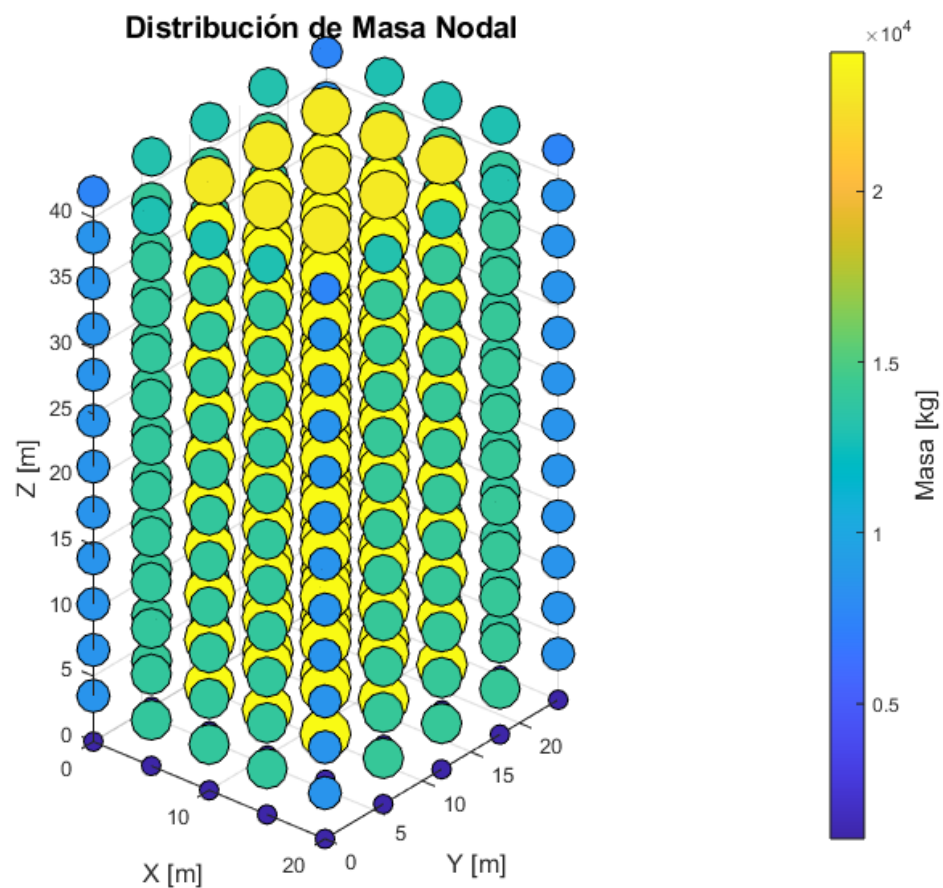


Figura 2.3: Distribución de masas nodales traslacionales del modelo estructural, resultado del proceso de tributación de peso propio según columnas, vigas y losas. El color está relacionado a la magnitud de la masa concentrada en cada nodo.

Finalmente, se construye una matriz de conectividad que relaciona cada elemento —columna o viga— con sus nodos extremos. Esta matriz es fundamental para el ensamblaje posterior de las matrices globales, ya que vincula los grados de libertad de la matriz global con los correspondientes a cada elemento. El script genera

las columnas y vigas en la malla ortogonal.

El modelo estructural generado puede visualizarse directamente. En la Figura 2.4 se observan vigas, columnas y losas distribuidas a lo largo de los ejes X , Y y Z .

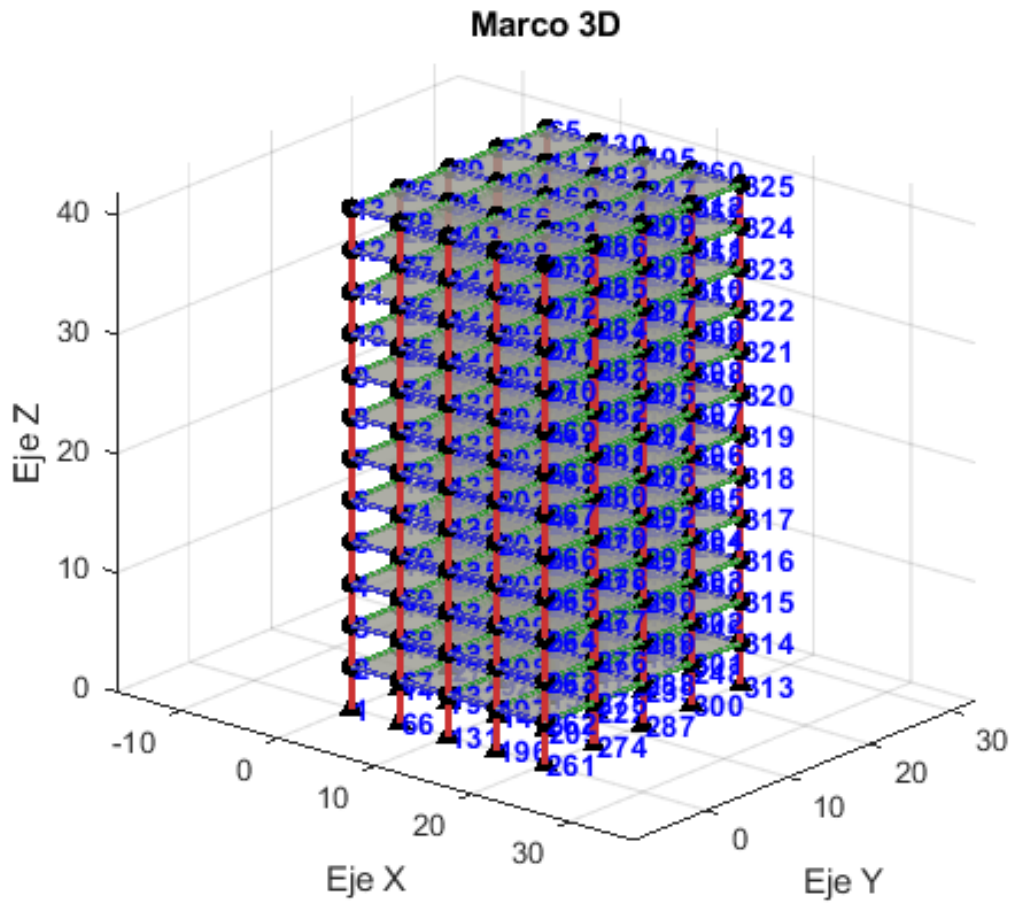


Figura 2.4: Marco en dos direcciones con vigas, columnas y losas. Incluye nodos numerados.

2.1.2. DIAFRAGMA RÍGIDO (marco3d_editable.m)

Dentro de este mismo script de preprocesamiento se implementa una rutina especial para modelar el comportamiento del diafragma rígido en cada nivel del edificio. La simplificación supone que la losa de cada piso es infinitamente rígida en su plano, con lo cual, los nudos de un mismo nivel quedan esclavizados cinemáticamente como un cuerpo rígido, es decir, sus desplazamientos horizontales quedan descritos mediante dos traslaciones y una rotación alrededor del eje vertical.

La aplicación del diafragma rígido reduce el número de grados de libertad activos del sistema y los concentra en el centro de masa (Villafuerte Lujano y Fernández-Dávila Gonzales, 2024). Cada nivel tiene sus propios grados de libertad maestros. Esta condensación cinemática mejora la eficiencia computacional del análisis al reducir el tamaño del sistema sin sacrificar la precisión en la representación del comportamiento estructural global, dado que la hipótesis del diafragma rígido en general es válida para edificios con sistemas de piso de concreto reforzado (Villafuerte Lujano y Fernández-Dávila Gonzales, 2024).

Los resultados generados por este script se presentan en el workspace de MATLAB y permiten verificar paso a paso la construcción y validación del modelo. A continuación se muestra el formato de salida utilizado, independientemente de la geometría introducida.

```
>> marco3d_editable
```

```
Creando columnas...
```

```
    N columnas creadas
```

```
Creando vigas en dirección Y...
```

```
    N vigas en Y creadas
```

```
Creando vigas en dirección X...
```

```
    N vigas en X creadas
```

```
RESUMEN DE ELEMENTOS
```

```
=====
```

```
Columnas:      N
```

```
Vigas Y:       N
```

```
Vigas X:       N
```

```
TOTAL:         N elementos
```

```
=====
```

Clasificando elementos por tipo...

N columnas clasificadas

N vigas clasificadas

Graficando losas...

N losas graficadas

Asignando restricciones de frontera...

N nodos restringidos (base)

N nodos libres

TOOLTIPS INTERACTIVOS ACTIVADOS

=====

Haz clic en cualquier nodo o barra

Aparecera informacion detallada

=====

Validando consistencia del modelo...

N nodos (coincide con nN)

N GDL totales

Construyendo matriz de conectividad LaGe...

LaGe: N x N

Validando indices de nodos en LaGe...

Indices de nodos validos

Total de nodos disponibles: N

```
Ensamblando matriz de masa...
  Matriz inicializada: N x N (sparse)
  Agregando masa de columnas...
  Agregando masa de vigas...
  Agregando inercia rotacional...
  Agregando masa de losas...
  Verificacion de consistencia satisfactoria
  Masa nodal calculada
```

```
Generando grafica de distribucion de masas...
  Grafica generada
```

```
Implementando diafragmas rigidos...
  Nivel i: Nodo master = N, Slaves = N
  Diafragmas definidos
```

2.1.3. MATRICES GLOBALES (mrigidez.m)

En este script se construyen las matrices globales de rigidez K_{global} y de masa M_{global} del sistema estructural completo. El proceso comienza con la inicialización de matrices dispersas de tamaño igual al número total de grados de libertad del modelo sin ninguna restricción. El uso de matrices dispersas mejora la eficiencia computacional, dado que las matrices de rigidez de sistemas estructurales presentan una gran proporción de elementos nulos: cada nodo solo está conectado directamente con un número limitado de nodos vecinos, correspondientes a los extremos de los elementos que confluyen en él (Dao et al., 2022; McGuire et al., 2000).

Para cada elemento tipo barra (columnas y vigas) se calcula su matriz de rigidez local en un sistema de coordenadas alineado con su eje longitudinal. Esta matriz se construye siguiendo las formulaciones clásicas de la teoría de vigas de Euler-Bernoulli, la cual considera la deformación por flexión alrededor de los dos ejes principales, la deformación axial y la torsión (Ghavami, 2015; McGuire et al., 2000). Los términos de la

matriz local incorporan las propiedades geométricas de cada elemento (área, momentos de inercia y constante de torsión), sus propiedades mecánicas (módulos de elasticidad y cortante) y su longitud.

Una vez obtenida la matriz de rigidez local, se realiza la transformación al sistema de coordenadas global mediante una matriz de rotación T de tamaño 12×12 (McGuire et al., 2000). Esta matriz considera la orientación espacial del elemento, definida por el vector que une sus nodos extremos. La matriz de rigidez global del elemento se obtiene mediante la expresión

$$K_{\text{global, elemento}} = T^T K_{\text{local}} T \quad (2.1)$$

Este proceso garantiza que las contribuciones de rigidez de cada elemento queden correctamente orientadas respecto al sistema de referencia global del edificio.

El ensamblaje de la matriz global de rigidez se realiza mediante superposición directa. Para cada elemento, los términos de su matriz global de tamaño 12×12 se suman en las posiciones correspondientes de la matriz K_{global} , identificadas mediante los grados de libertad de sus nodos extremos (Celigüeta-Lizarza, 2011). Esta operación se realiza gracias a la numeración sistemática de cada grado de libertad, donde cada nodo tiene asociados seis grados de libertad consecutivos (tres traslaciones y tres rotaciones).

Una vez ensambladas K_{global} y M_{global} , se ejecuta una rutina de depuración que identifica y elimina grados de libertad problemáticos. La rutina analiza las filas y columnas de la matriz de rigidez para detectar grados de libertad desconectados —aquellos cuya fila y columna son completamente nulas— o aquellos asociados a modos rígidos con rigidez extremadamente baja. La depuración reduce el sistema a los grados de libertad efectivos y mejora el condicionamiento numérico, evitando problemas de singularidad en las etapas siguientes (Gloyne, 1964; McGuire et al., 2000).

A continuación se aplica la transformación asociada al diafragma rígido mediante

$$K_{\text{con diafragma}} = T^T K_{\text{global}} T \quad (2.2)$$

y

$$\mathcal{M}_{\text{con diafragma}} = \mathbf{T}^T \mathcal{M}_{\text{global}} \mathbf{T} \quad (2.3)$$

Esta transformación condensa los grados de libertad horizontales de cada nivel en los grados de libertad maestros correspondientes. El patrón de banda característico de la matriz ensamblada confirma la correcta conectividad del sistema (Figura 2.5): cada banda paralela a la diagonal principal representa el acoplamiento entre grados de libertad de nodos conectados por elementos estructurales.

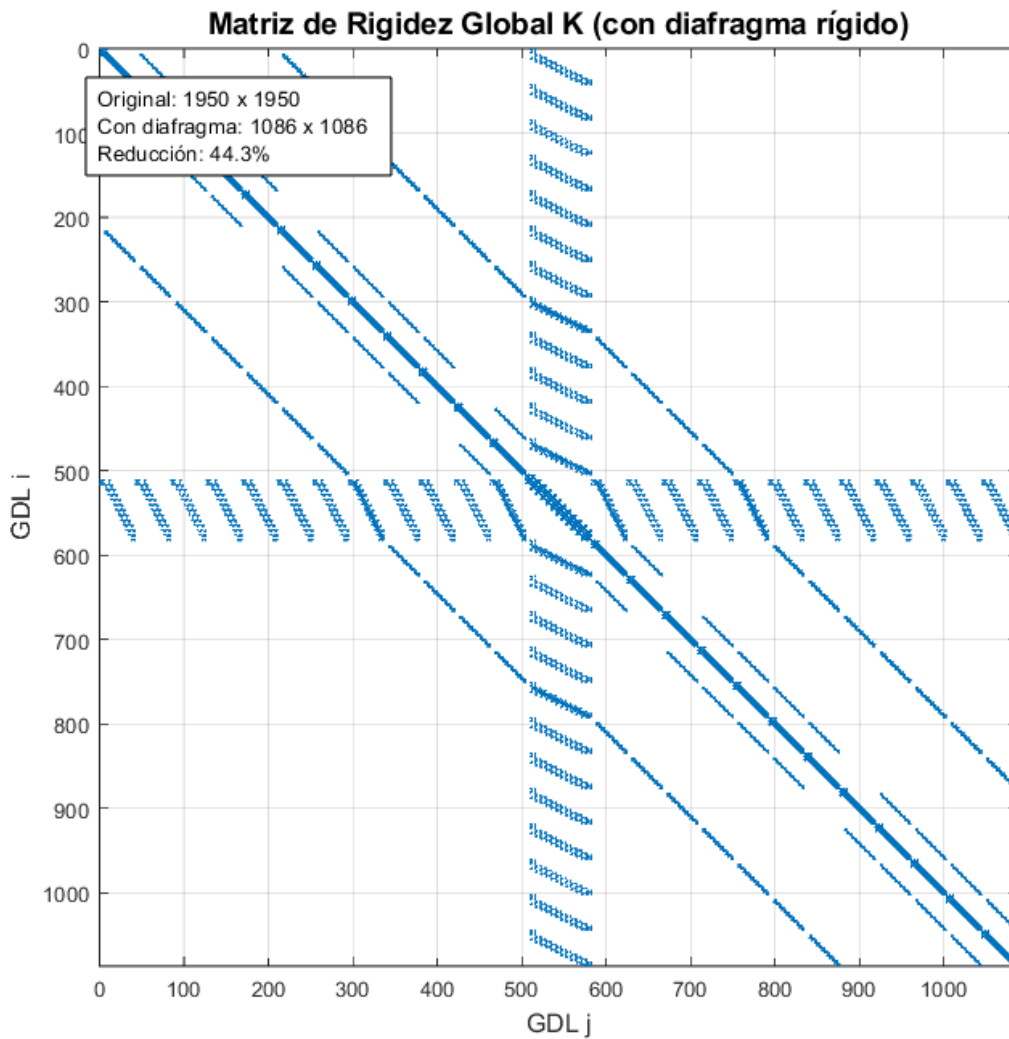


Figura 2.5: Patrón de banda de la matriz de rigidez global K tras la aplicación del diafragma rígido. La reducción de grados de libertad refleja la condensación cinemática de los desplazamientos horizontales de cada nivel en los grados de libertad maestros.

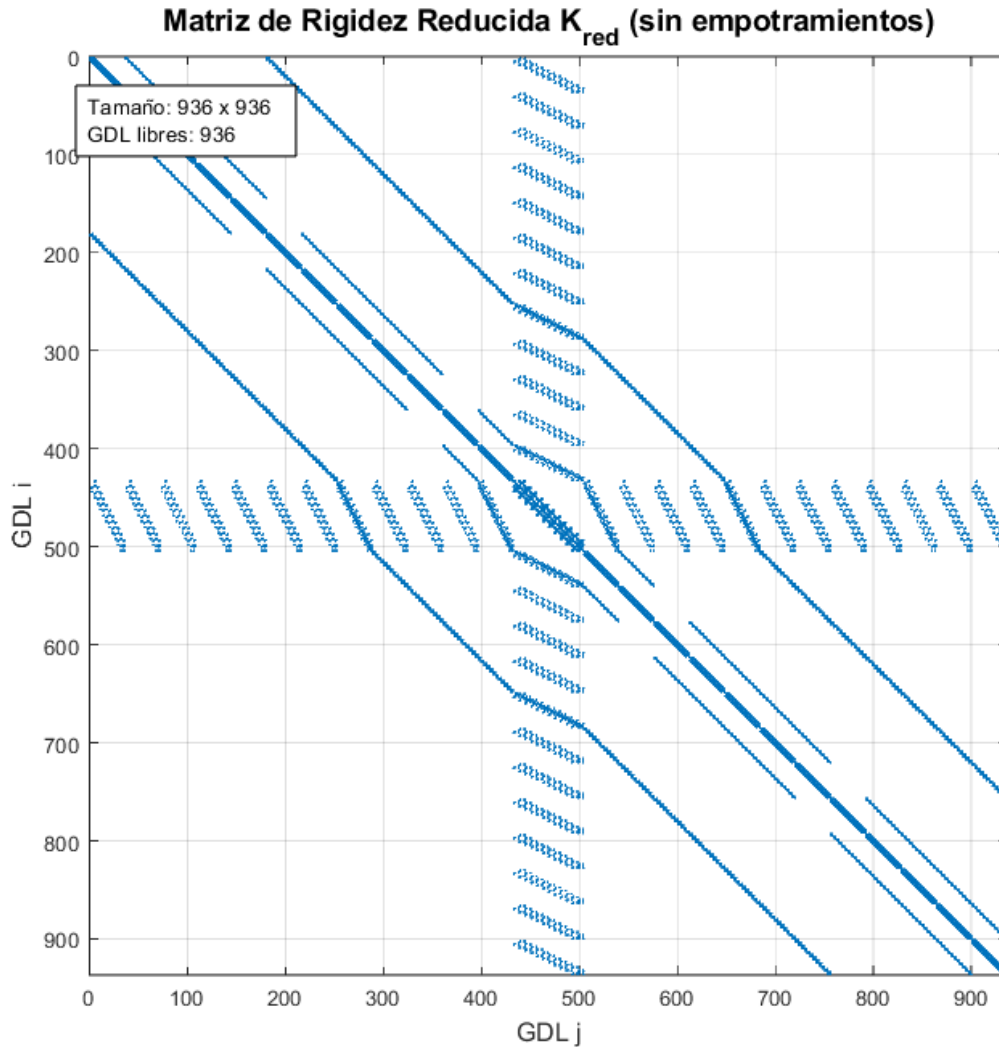


Figura 2.6: Patrón de banda de la matriz de rigidez reducida K_{red} tras la eliminación de los grados de libertad restringidos en la base. Los grados de libertad libres resultantes son los que participan en los análisis dinámicos subsecuentes.

2.1.4. CONDICIONES DE FRONTERA (`mrigidez.m`)

Dentro del mismo script de ensamblaje de la matriz de rigidez se implementa la aplicación de las condiciones de frontera que representan los apoyos de la estructura. Se identifican todos los nodos ubicados en la base del edificio —aquellos con coordenada $z = 0$ —, que corresponden a la interfaz entre la superestructura y el sistema de cimentación. Para el análisis con base empotrada se restringen los seis grados de libertad de cada nodo de la base, lo que representa una condición de empotramiento perfecto que impide cualquier desplazamiento o rotación en la base.

La aplicación de las condiciones de frontera se realiza mediante la identificación de los índices de los grados de libertad restringidos y libres. Los grados de libertad restringidos corresponden a todos los nodos de la base; los libres son los restantes, que participan activamente en la respuesta estructural. Con esta clasificación se generan dos vectores de índices: `gdl_restringidos` y `gdl_libres`. Las matrices reducidas de rigidez y de masa se obtienen extrayendo las filas y columnas correspondientes a los grados de libertad libres. Estas matrices constituyen el sistema dinámico efectivo de la estructura que se resuelve en los análisis estático, modal y espectral subsecuentes. La reducción del sistema mediante la eliminación de los grados de libertad restringidos es matemáticamente equivalente a la condensación estática de Guyan y garantiza que la solución satisface exactamente las condiciones de frontera impuestas (Wada, 1965).

A partir de este punto el proceso tiene tres salidas disponibles según el tipo de análisis que se realice. Esta decisión se representa en el rombo de decisión del diagrama general de la Figura 2.2, con las posibles opciones de análisis estático, dinámico o de interacción suelo-estructura. Con esto es posible ejecutar cualquiera de los tres análisis de manera independiente, manteniendo el mismo modelo geométrico base generado en los módulos de preprocesamiento y ensamblaje de matrices.

Los resultados generados por este script se presentan en el workspace de MATLAB y permiten verificar la formulación, reducción y validación de las matrices estructurales. A continuación se muestra el formato de salida utilizado, independientemente de la geometría introducida.

```
>> mrigidez
```

```
ENSAMBLAJE DE MATRIZ DE RIGIDEZ GLOBAL
```

```
=====
```

```
Información del modelo:
```

```
Nodos:      N
```

```
Barras:     N
```

```
GDL:        N
```

Inicializando matrices...

Matriz K: N x N (sparse)

Ensamblando matriz de rigidez...

Columnas ensambladas

Vigas ensambladas

Total de elementos ensamblados

Verificando propiedades de la matriz K...

Simetría: verificada

Elementos no cero: matriz dispersa

Diagonal: elementos positivos

APLICANDO DIAFRAGMA RÍGIDO (Master-Slave)

=====

Identificando GDL dependientes...

GDL dependientes: N

GDL originales: N

Construyendo matriz de transformación...

Matriz T generada

Transformando matrices K y M...

Matrices transformadas

Reducción de GDL aplicada

Diafragmas aplicados:

Pisos con diafragma rígido

GDL eliminados

GDL restantes

Aplicando condiciones de frontera...

GDL restringidos (empotramientos)

GDL libres

Matrices reducidas generadas:

Tamaño: $N \times N$

K_{red} : matriz dispersa

M_{red} : matriz dispersa

Verificando definición positiva de K_{red} ...

Autovalores positivos

Número de condición aceptable

K_{red} definida positiva

Vector de fuerzas externas inicializado

Tamaño: $N \times 1$

Generando graficas de estructura de matrices...

Gráficas generadas

RESUMEN DE MATRICES GENERADAS

=====

Diafragma rígido:

Aplicado
Pisos
GDL eliminados

Matriz de Rigidez:

Kglobal (con diafragma)
K_red (sin empotramientos)

Matriz de Masa:

Mglobal (con diafragma)
M_red (sin empotramientos)

Vector de Fuerzas:

F_ext (inicializado en ceros)

GDL:

Total (con diafragma)
Restringidos
Libres

=====

Ensamblaje completado

2.2. DESARROLLO DEL CÓDIGO

En esta sección se explica la forma en que cada algoritmo correspondiente a cada análisis es ejecutado, desde la reducción del sistema tras la aplicación de las condiciones de frontera hasta la incorporación del suelo en el análisis de interacción suelo-estructura. Los tres tipos de análisis (estático, dinámico e interacción

suelo-estructura) utilizan las mismas matrices globales de rigidez y masa generadas en el preprocesamiento, modificando únicamente las condiciones de frontera según el análisis: empotramientos rígidos para el análisis convencional, o rigideces equivalentes del suelo para el análisis con ISE.

2.2.1. ANÁLISIS ESTÁTICO

El análisis estático se realiza a partir del sistema reducido, una vez aplicadas las condiciones de frontera, de forma que solo participan los grados de libertad activos. Este procedimiento sigue la formulación del método de la matriz de rigidez, representado por la ecuación

$$\mathbf{K}_{ff} \mathbf{U}_f = \mathbf{F}_f \quad (2.4)$$

La ecuación 2.4 se resuelve sin introducir no linealidades ni ajustes posteriores adicionales (McGuire et al., 2000).

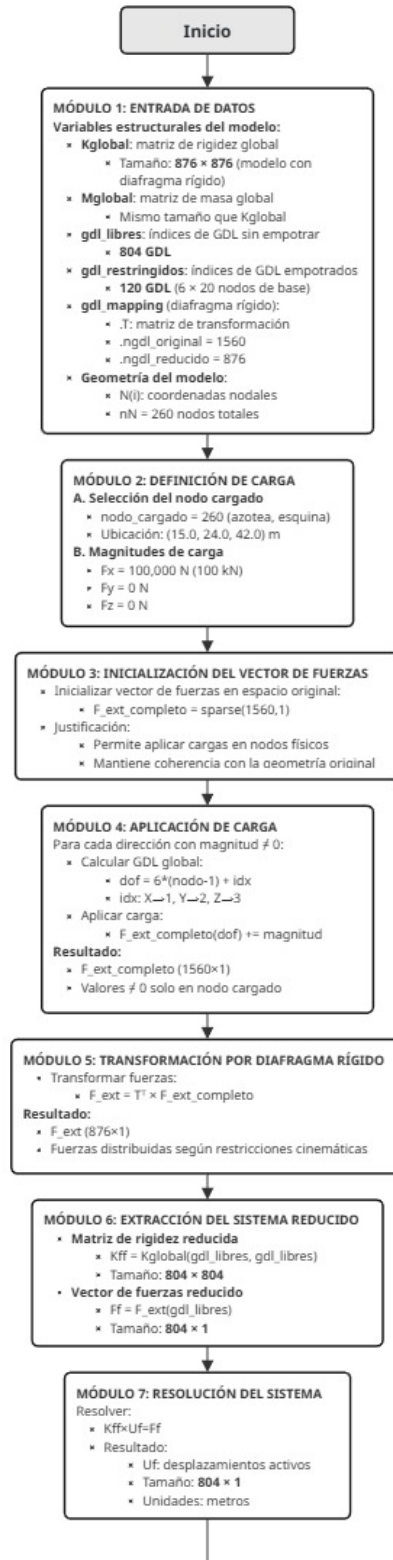


Figura 2.7: Diagrama de decisiones del análisis estático. Continúa en la siguiente página.

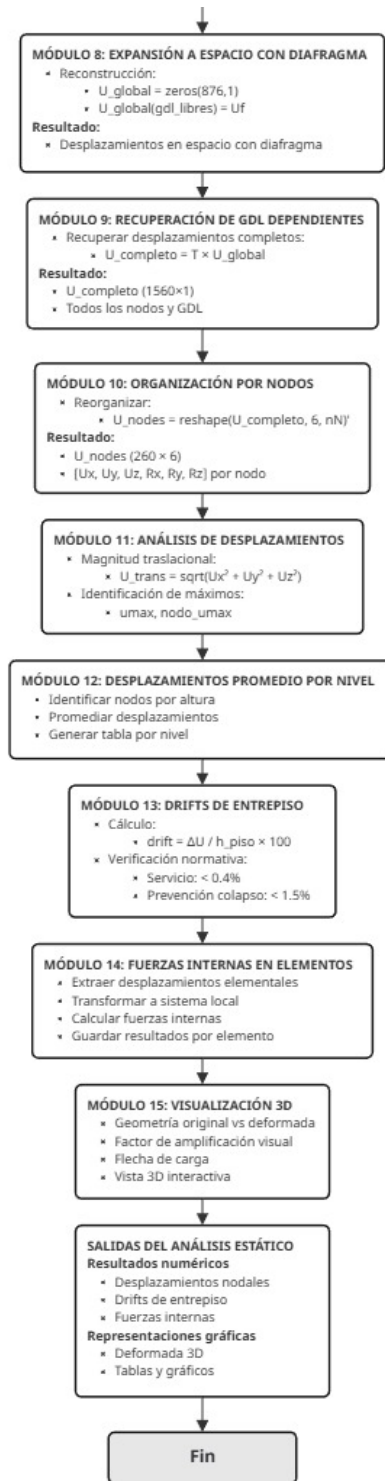


Figura 2.8: Continuación del diagrama de decisiones del Análisis Estático.

El proceso computacional llevado a cabo para este análisis se puede observar en las Figuras 2.7 y 2.8 y se divide en los siguientes módulos secuenciales.

ENTRADA DE DATOS

Las matrices de rigidez global K_{global} y de masa M_{global} constituyen la base del análisis. En esta etapa también se cargan los vectores de índice que identifican los grados de libertad libres (`gdl_libres`) y los restringidos (`gdl_restringidos`), así como la matriz de transformación del diafragma rígido (`gdl_mapping`), que relaciona el sistema original de grados de libertad con el sistema condensado. Finalmente se carga la matriz de coordenadas nodales (x,y,z) , que contiene las posiciones en el espacio de todos los nodos del modelo y cuya información es útil para la visualización posterior de los resultados.

DEFINICIÓN DE LA CARGA

Aquí se define el nodo donde se aplicará la carga para el análisis, así como las magnitudes de las fuerzas en cada dirección. Para este caso se selecciona el último nodo generado, que corresponde a la esquina superior del edificio en la azotea. Este nodo fue elegido porque en geometrías regulares representa la posición más desfavorable para una carga lateral: se encuentra en el extremo superior de la estructura y alejado del centro de rigidez, lo que maximiza tanto el desplazamiento traslacional como el efecto torsional. Las magnitudes de carga pueden especificarse en las direcciones X , Y y Z , lo que permite evaluar la respuesta lateral de la estructura en cualquiera de sus direcciones principales.

INICIALIZACIÓN DEL VECTOR DE FUERZAS

Se inicializa un vector de fuerzas con longitud correspondiente al sistema original sin diafragma, con todos los elementos iguales a cero, mediante la función `sparse` de MATLAB, a fin de mantener la eficiencia en el manejo de memoria computacional. Este vector representa las cargas externas aplicadas en todos los grados de libertad del modelo. La inicialización con formato disperso es importante porque únicamente un número reducido de grados de libertad tendrá fuerzas aplicadas, mientras que los restantes permanecerán con valores

nulos.

APLICACIÓN DE CARGA

Para cada dirección con magnitud diferente de cero, se calcula el índice del grado de libertad correspondiente en el vector global. El nodo en el que se aplica la fuerza tiene asociados seis grados de libertad consecutivos, numerados como $6 \times (\text{nodo} - 1) + \text{direccion}$, donde la dirección toma valores del 1 al 6 según los componentes $U_x, U_y, U_z, R_x, R_y, R_z$. Se asigna la magnitud de carga en la posición correspondiente del vector. El resultado es un vector disperso donde solo la posición asociada al grado de libertad cargado tiene un valor diferente de cero.

TRANSFORMACIÓN POR DIAFRAGMA RÍGIDO

El vector de fuerzas definido en el sistema original debe transformarse al sistema condensado por diafragma rígido mediante la operación $\mathbf{F}_{\text{ext}} = \mathbf{T}^T \times \mathbf{F}_{\text{ext,completo}}$, donde \mathbf{T} es la matriz de transformación construida en el módulo de preprocesamiento (`marco3d_editable.m`). Esta operación matricial distribuye la fuerza aplicada entre los grados de libertad maestros del diafragma correspondiente, conforme a las relaciones cinemáticas establecidas. El vector resultante contiene las fuerzas distribuidas de forma coherente con la hipótesis de diafragma rígido: la carga horizontal aplicada en un nodo se transmite al nodo maestro del nivel y genera un momento torsional proporcional a la excentricidad entre el punto de aplicación y el centro del diafragma.

EXTRACCIÓN DEL SISTEMA REDUCIDO

En esta etapa se extraen las submatrices y subvectores correspondientes a los grados de libertad libres. La matriz de rigidez reducida \mathbf{K}_{ff} se obtiene mediante $\mathbf{K}_{ff} = \mathbf{K}_{\text{global}}(\text{gdl_libres}, \text{gdl_libres})$, que da como resultado una matriz que contiene únicamente los acoplamientos entre grados de libertad activos. El vector de fuerzas reducido \mathbf{F}_f se obtiene de manera análoga mediante $\mathbf{F}_f = \mathbf{F}_{\text{ext}}(\text{gdl_libres})$. Esta extracción representa la aplicación de las condiciones de empotramiento en la base: los desplazamientos de

los nodos de la base se imponen como nulos y los grados de libertad correspondientes se eliminan del sistema de ecuaciones.

RESOLUCIÓN DEL SISTEMA

En este módulo se resuelve el sistema lineal $\mathbf{K}_{ff}\mathbf{U}_f = \mathbf{F}_f$ para obtener el vector de desplazamientos \mathbf{U}_f en los grados de libertad libres. En MATLAB esta operación se ejecuta eficientemente con el operador de división izquierda $\mathbf{U}_f = \mathbf{K}_{ff} \setminus \mathbf{F}_f$, que implementa algoritmos optimizados de factorización matricial y aprovecha la estructura dispersa y simétrica de la matriz de rigidez. El vector resultante \mathbf{U}_f contiene los desplazamientos traslacionales y rotacionales de todos los nodos activos en el sistema de referencia global, y representa la deformada estática del marco bajo la carga aplicada.

EXPANSIÓN A ESPACIO CON DIAFRAGMA

Los desplazamientos \mathbf{U}_f se reintegran en un vector global $\mathbf{U}_{\text{global}}$ que incluye también los grados de libertad restringidos. Se inicializa el vector $\mathbf{U}_{\text{global}}$ con todos sus componentes iguales a cero. Los desplazamientos calculados se insertan en las posiciones correspondientes a los grados de libertad libres: $\mathbf{U}_{\text{global}}(\text{gdl_libres}) = \mathbf{U}_f$. Las posiciones correspondientes a los grados de libertad restringidos mantienen su valor nulo, lo que representa correctamente la condición de empotramiento en la base.

RECUPERACIÓN DE GDL DEPENDIENTES

Para obtener los desplazamientos en el sistema original sin diafragma se aplica la transformación inversa $\mathbf{U}_{\text{completo}} = \mathbf{T} \times \mathbf{U}_{\text{global}}$. Con esta operación se distribuyen los desplazamientos de los nodos maestros de cada diafragma hacia los nodos dependientes, o esclavos, del nivel correspondiente, aplicando las relaciones cinemáticas del diafragma rígido. El vector $\mathbf{U}_{\text{completo}}$ resultante contiene los desplazamientos de todos los grados de libertad del modelo original.

ORGANIZACIÓN POR NODOS

Los desplazamientos del vector $\mathbf{U}_{\text{completo}}$ se reorganizan en la matriz U_{nodos} , donde cada fila representa un nodo y las columnas corresponden a los seis grados de libertad $(U_x, U_y, U_z, R_x, R_y, R_z)$. Esta matriz facilita el acceso a los desplazamientos de nodos específicos y simplifica el procesamiento posterior de los resultados. Por ejemplo, $U_{\text{nodos}}(:, 1)$ contiene todos los desplazamientos en la dirección X de todos los nodos del modelo, a partir de los cuales es posible calcular magnitudes derivadas como desplazamientos totales, identificación de máximos y análisis por nivel.

ANÁLISIS DE DESPLAZAMIENTOS

En esta sección se calculan las magnitudes de los desplazamientos traslacionales en cada nodo mediante $U_{\text{trans}} = \sqrt{U_x^2 + U_y^2 + U_z^2}$. Se identifica el desplazamiento máximo y el nodo donde ocurre mediante las funciones de búsqueda de máximo de MATLAB. Para una carga horizontal aplicada en la azotea, el desplazamiento máximo debe ocurrir en los niveles superiores y ser mayor en la dirección de aplicación de la carga, condición que permite verificar la coherencia del comportamiento estructural con la física del problema estático.

DESPLAZAMIENTOS PROMEDIO POR NIVEL

Para cada nivel del edificio se identifican los nodos con la misma coordenada Z y se promedian sus desplazamientos horizontales para obtener un desplazamiento representativo del piso. El resultado es una tabla cuyas filas corresponden a los niveles de la estructura, de la planta baja a la azotea, con los desplazamientos promedio en X , Y y Z . Estos valores permiten visualizar la deformada global de la estructura y constituyen la base para el cálculo de distorsiones de entrepiso. El patrón de desplazamientos debe mostrar valores crecientes desde la base hacia la azotea. En la Figura 2.9 se observa la deformada tridimensional, con la opción de amplificarla gráficamente para mejor observación del comportamiento estructural. La amplificación puede ir de $1 \times$ hasta $500 \times$ y es únicamente con fines visuales, sin afectar los cálculos posteriores.

DESPLAZAMIENTOS DEL ENTREPISO

Las distorsiones de entrepiso se calculan para cada nivel como el desplazamiento relativo entre pisos consecutivos dividido entre la altura del entrepiso, expresado como porcentaje:

$$\Delta_{\text{entrepiso}} = \frac{U_{\text{superior}} - U_{\text{inferior}}}{b} \times 100 \% \quad (2.5)$$

Este cálculo se realiza para cada dirección principal X y Y , utilizando los desplazamientos promedio de cada nivel calculados en la sección anterior. Las Normas Técnicas Complementarias para Diseño por Sismo de la CDMX establecen límites de distorsión de entrepiso de 0.4 % para estado de servicio y 1.5 % para estado de colapso en estructuras de concreto con $Q = 2$.

FUERZAS INTERNAS EN LOS ELEMENTOS

Los desplazamientos nodales extremos de cada elemento estructural se extraen del vector $\mathbf{U}_{\text{completo}}$. Estos desplazamientos globales se transforman al sistema de coordenadas locales del elemento mediante la matriz de rotación \mathbf{T} . Con los resultados en el sistema local, las fuerzas internas se calculan como

$$\mathbf{F}_{\text{local}} = \mathbf{K}_{\text{local}} \times \mathbf{U}_{\text{local}} \quad (2.6)$$

El vector de fuerzas internas contiene las componentes axiales, cortantes, momentos flectores y momentos de torsión en cada extremo del elemento. Las fuerzas internas calculadas permiten verificar la validez del modelo: la suma de las reacciones en los apoyos debe igualar a las cargas aplicadas, y las fuerzas internas deben satisfacer el equilibrio en cada nodo de la estructura.

VISUALIZACIÓN 3D

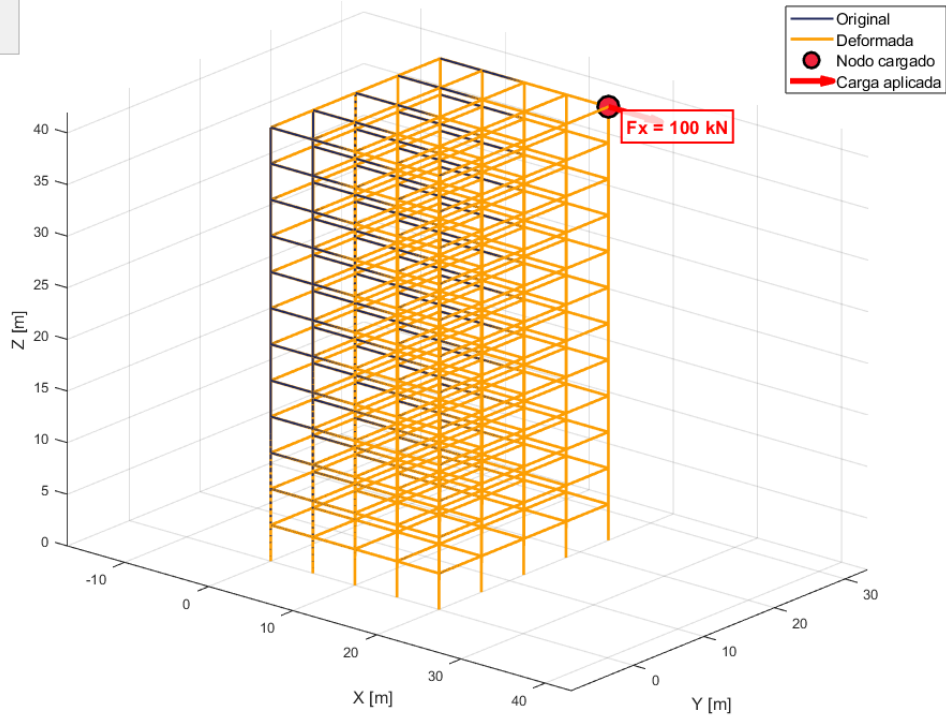
Finalmente, se genera una representación gráfica tridimensional que superpone la geometría original del marco con la deformada. La geometría original se dibuja con líneas de color negro y la deformada con líneas

de color naranja. La representación incluye la indicación de la fuerza aplicada con su dirección y magnitud. El modelo 3D permite verificar visualmente la coherencia del comportamiento estructural: el edificio debe desplazarse de manera predominante en la dirección de la carga aplicada, con desplazamientos crecientes hacia los niveles superiores.

Control de Amplificación
Factor: 1x

1x 500x

Marco 3D - Deformada (Factor escala: 1x)



Control de Amplificación
Factor: 500x

1x 500x

Marco 3D - Deformada (Factor escala: 500x)

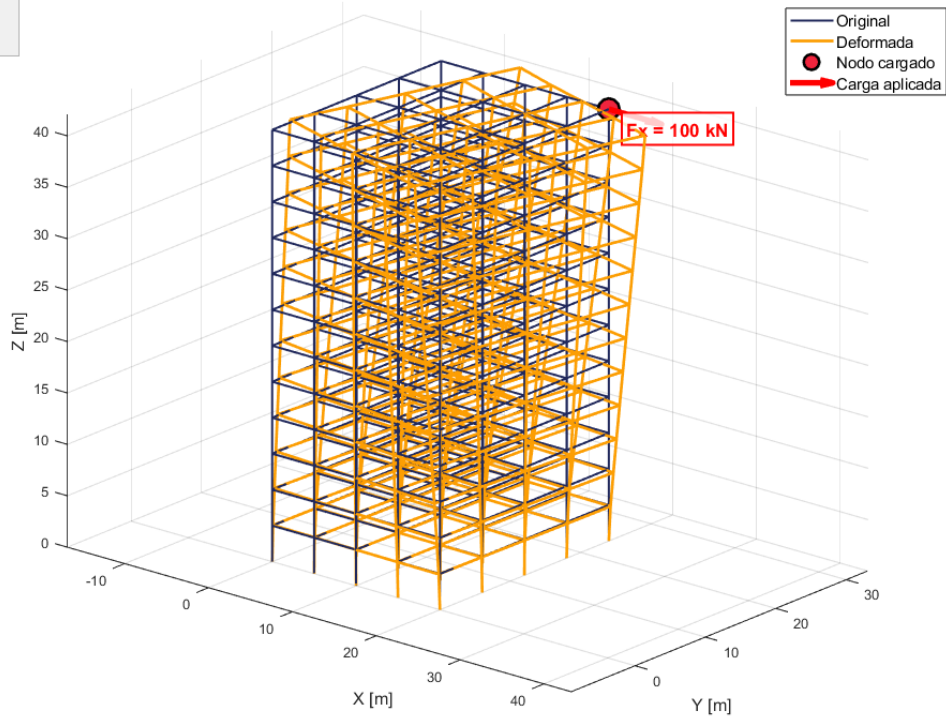


Figura 2.9: Vista tridimensional de la deformada bajo carga puntual estática. La imagen superior corresponde a escala real ($1 \times$) y la inferior a una amplificación de $500 \times$, únicamente con fines visuales.

SALIDAS DEL ANÁLISIS ESTÁTICO

El script genera los siguientes resultados: desplazamientos nodales completos, desplazamientos promedio por nivel en las tres direcciones, distorsiones de entrepiso por cada piso y dirección, desplazamiento máximo del edificio y el nodo donde ocurre, y fuerzas internas en todos los elementos estructurales. Las representaciones gráficas incluyen la deformada tridimensional superpuesta, tablas de desplazamientos por nivel y gráficas de desplazamiento en función de la altura del edificio. Estos resultados se presentan de la siguiente manera.

```
>> analisis_estatico
```

ANÁLISIS ESTÁTICO - CARGA PUNTUAL

```
=====
```

Información del modelo:

Nodos totales

GDL totales

Niveles

Nodo seleccionado para carga puntual:

Nodo

Posición: X, Y, Z

(Esquina posterior-derecha de azotea)

Condiciones de frontera:

GDL restringidos (empotramiento base)

GDL libres

Definiendo carga puntual...

Inicializando vector de fuerzas

Aplicando carga en nodo

Transformando fuerzas al espacio con diafragma...

Fuerzas transformadas

Resumen de carga:

Nodo cargado

Ubicación

Fuerza en X

Fuerza en Y

Fuerza en Z

Resolviendo sistema de ecuaciones...

Sistema resuelto

Metodo: Eliminación directa (operador backslash)

Tamaño del sistema

Reconstruyendo desplazamientos completos...

Desplazamientos reconstruidos

Analizando desplazamientos...

Desplazamiento máximo:

Magnitud

Nodo

Ubicación

Nodo cargado:

Ux

Uy

Uz

Desplazamientos por nivel:

Nivel	Altura	Ux	Uy	Uz
-------	--------	----	----	----

Niveles procesados

Desplazamientos del entrepiso:

Piso	Drift_X	Drift_Y	Drift_Z
------	---------	---------	---------

Pisos procesados

Calculando fuerzas internas en elementos...

Elementos procesados

Generando visualización...

Visualizacion generada

RESUMEN DE ANÁLISIS ESTÁTICO

=====

Carga aplicada:

Nodo (azotea)

Fx

Fy

Fz

Desplazamiento maximo:

Magnitud

Nodo

Desplazamiento en nodo cargado:

Ux

Uy

Uz

Drift maximo:

Direccion principal

=====

Análisis estático completado

2.2.2. ANÁLISIS DINÁMICO

El análisis dinámico se inicia con la resolución del problema modal, formulado a partir de las submatrices previamente calculadas y reducidas de masa y rigidez. La ecuación

$$(K_{ff} - \omega_i^2 M_{ff}) \varphi_i = 0 \quad (2.7)$$

se resuelve siguiendo los criterios de dinámica estructural (Chopra, 2007). Los modos obtenidos se depuran para eliminar valores no físicos asociados a grados de libertad mal condicionados (Ángel et al., 2024).

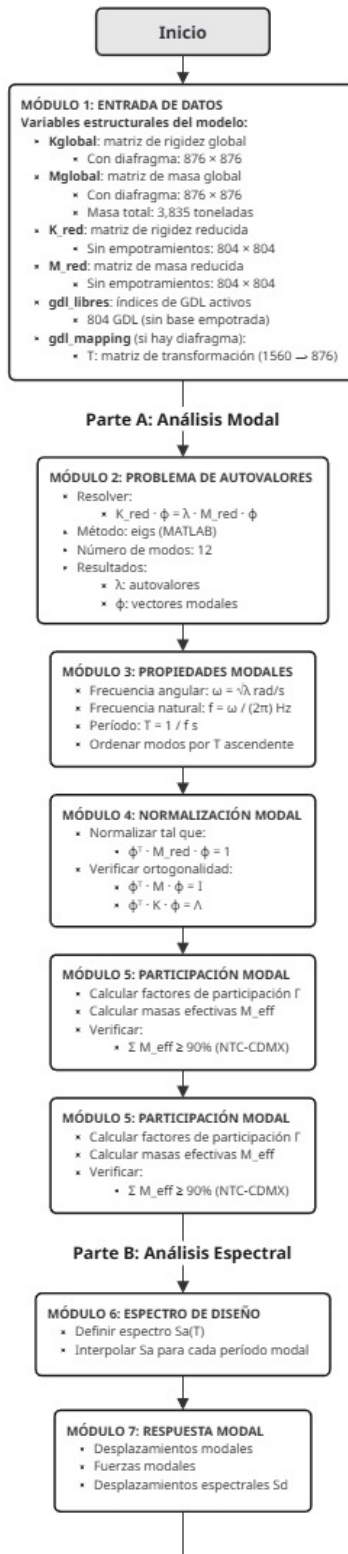


Figura 2.10: Diagrama del Análisis Dinámico que incluye análisis modal y espectral. Continúa en la siguiente página.

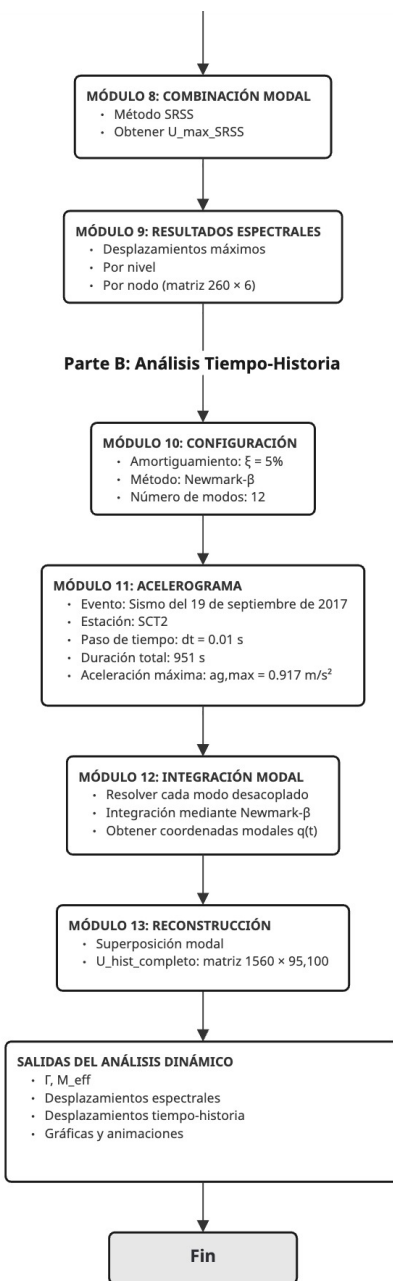


Figura 2.11: Continuación del diagrama de decisiones del Análisis Dinámico.

El diagrama de decisiones del análisis dinámico se puede ver en las Figuras 2.10 y 2.11, dividido principalmente en dos partes: análisis modal y análisis modal espectral. A continuación se describen los componentes de cada uno.

ANÁLISIS MODAL - ENTRADA DE DATOS

El script de análisis dinámico utiliza como base, al igual que el análisis estático, los resultados generados en los módulos de preprocesamiento y ensamblaje de matrices. Se incorpora también la matriz de masa global M_{global} y las matrices reducidas de rigidez y de masa calculadas tras la aplicación de las condiciones de frontera. Los vectores `gdl_libres` identifican los grados de libertad activos que participarán en el análisis modal. La matriz `gdl_mapping` permite transformar entre el sistema con diafragma y el sistema original, ambos necesarios para la obtención de los modos de vibrar.

ANÁLISIS MODAL - PROBLEMA DE AUTOVALORES

El problema de autovalores generalizado se plantea como

$$(K_{\text{red}} - \lambda M_{\text{red}})\varphi = 0 \quad (2.8)$$

y se resuelve mediante la función `eigs` de MATLAB, que emplea métodos numéricos iterativos especializados adecuados para matrices dispersas de gran tamaño (McGuire et al., 2000; Villafuerte Lujano y Fernández-Dávila Gonzales, 2024). Se calculan los 12 modos con menores frecuencias mediante la opción `smallestabs`, que extrae los autovalores de menor magnitud. Esta elección es adecuada para el análisis sísmico, ya que los modos de baja frecuencia —con períodos largos— dominan típicamente la respuesta estructural (Villafuerte Lujano y Fernández-Dávila Gonzales, 2024). El algoritmo produce dos matrices: Λ , diagonal con los autovalores $\lambda_i = \omega_i^2$, y Φ , cuyas columnas contienen los autovectores φ_i , cada uno representando la forma modal asociada a una frecuencia natural.

ANÁLISIS MODAL - PROPIEDADES MODALES

Los autovalores obtenidos se convierten en propiedades dinámicas físicas. La frecuencia angular ω_i se calcula como $\omega_i = \sqrt{\lambda_i}$, expresada en radianes por segundo. La frecuencia natural f_i se obtiene como $f_i = \omega_i/(2\pi)$, en Hertz. El período natural es $T_i = 1/f_i$, en segundos. Los modos se ordenan automáticamente de mayor a menor período, de modo que el modo 1 corresponde al período fundamental —el mayor período natural de la estructura—. El valor del período fundamental puede verificarse mediante la fórmula empírica de las NTC-Diseño por Sismo:

$$T_a = 0.05 H^{3/4} \quad (2.9)$$

donde H es la altura total del edificio.

ANÁLISIS MODAL - NORMALIZACIÓN MODAL

Los autovectores obtenidos del problema de autovalores se normalizan respecto a la matriz de masa para garantizar la ortogonalidad modal. Para cada modo i , el factor de normalización es

$$\alpha_i = \sqrt{\varphi_i^T \mathbf{M}_{\text{red}} \varphi_i}, \quad (2.10)$$

y el autovector normalizado es

$$\hat{\varphi}_i = \frac{\varphi_i}{\alpha_i}. \quad (2.11)$$

Con esta normalización se garantiza que

$$\hat{\varphi}_i^T \mathbf{M}_{\text{red}} \hat{\varphi}_i = 1 \quad \forall i, \quad (2.12)$$

condición conocida como normalización de masa unitaria. Este procedimiento garantiza que los modos naturales de vibración conforman un conjunto completo y desacoplado, lo que permite aplicar la técnica de superposición modal (Avilés López et al., 2016; Wada, 1965).

La ortogonalidad modal se verifica mediante

$$\mathbf{M}_{\text{modal}} = \Phi^T \mathbf{M}_{\text{red}} \Phi, \quad \mathbf{K}_{\text{modal}} = \Phi^T \mathbf{K}_{\text{red}} \Phi, \quad (2.13)$$

donde $\mathbf{M}_{\text{modal}}$ debe ser la matriz identidad y $\mathbf{K}_{\text{modal}}$ debe ser diagonal con los autovalores λ_i en la diagonal.

El error de ortogonalidad de masa se cuantifica mediante

$$\|\mathbf{M}_{\text{modal}} - \mathbf{I}\|_F, \quad (2.14)$$

donde \mathbf{I} es la matriz identidad y $\|\cdot\|_F$ es la norma de Frobenius (McGuire et al., 2000).

Para la rigidez se evalúa el error relativo

$$\frac{\|\mathbf{K}_{\text{modal}} - \text{diag}(\mathbf{K}_{\text{modal}})\|_F}{\|\mathbf{K}_{\text{modal}}\|_F}. \quad (2.15)$$

Valores de error inferiores a 10^{-10} para la matriz de masa y de 10^{-6} para la matriz de rigidez son aceptables y confirman que los modos calculados son ortogonales respecto a ambas matrices, validando el uso del método de superposición modal (McGuire et al., 2000).

ANÁLISIS MODAL - PARTICIPACIÓN MODAL

Los vectores de influencia sísmica \mathbf{r}_x , \mathbf{r}_y , \mathbf{r}_z se construyen en el espacio de grados de libertad libres del sistema reducido. Dado que la condensación por diafragma rígido reordena los índices de los grados de libertad, el tipo traslacional de cada grado de libertad no puede determinarse directamente desde el espacio reducido. Por ello, cada índice del espacio reducido se mapea al espacio original de $6 \cdot n_N$ grados de libertad

mediante la relación `gdl_mapping.independientes`, y el tipo de grado de libertad se identifica como $\text{mod}(\text{idx_original} - 1, 6) + 1$. Los valores 1, 2 y 3 corresponden a traslaciones en x , y y z , respectivamente; los valores 4, 5 y 6 corresponden a rotaciones y reciben componente nula en los tres vectores.

La masa total que se desplaza en cada dirección se calcula mediante la forma cuadrática

$$M_{\text{total},x} = \mathbf{r}_x^\top \mathbf{M}_{\text{red}} \mathbf{r}_x, \quad (2.16)$$

y de manera análoga para y y z . Esta expresión es equivalente a la definida en la NTC-Sismo 2020 y es necesaria porque la condensación cinemática por diafragma rígido produce una matriz de masa densa, con acoplamientos de inercia fuera de la diagonal:

$$\mathbf{M}_{\text{red}} = \mathbf{T}^\top \mathbf{M} \mathbf{T}. \quad (2.17)$$

Sumar únicamente la diagonal de una submatriz traslacional subestimaría la masa efectiva que se desplaza. El porcentaje de participación del modo i en la dirección x queda entonces como

$$\% \text{Part}_{xi} = \frac{M_{\text{eff},xi}}{M_{\text{total},x}} \times 100. \quad (2.18)$$

Las NTC requieren que la suma de los modos considerados represente al menos el 90 % de la masa total en cada dirección.

ANÁLISIS ESPECTRAL - ESPECTRO DE DISEÑO

La función de espectro de respuesta de aceleraciones $S_a(T)$ se implementa conforme a las Normas Técnicas Complementarias para Diseño por Sismo del 2017. El espectro se construye a través de la formulación por tramos establecida en la normativa, considerando los parámetros sísmicos que dependen de la zona de suelo donde se ubica la estructura.

El espectro de diseño empleado corresponde al espectro de aceleraciones obtenido del Sistema de Acciones Sísmicas de Diseño (SASID), para la Zona III - Lago de la Ciudad de México, con factor de comportamiento sísmico $Q = 2$. El archivo de entrada contiene 51 pares (T, S_a) con períodos entre 0.01 y 5.00 s y aceleraciones en Gal. El script interpola linealmente este espectro en los períodos modales obtenidos del análisis previo mediante la función `interp1` de MATLAB, sin reducción adicional por Q . Esta forma de operar el espectro es consistente con el supuesto de comportamiento elástico-lineal del modelo.

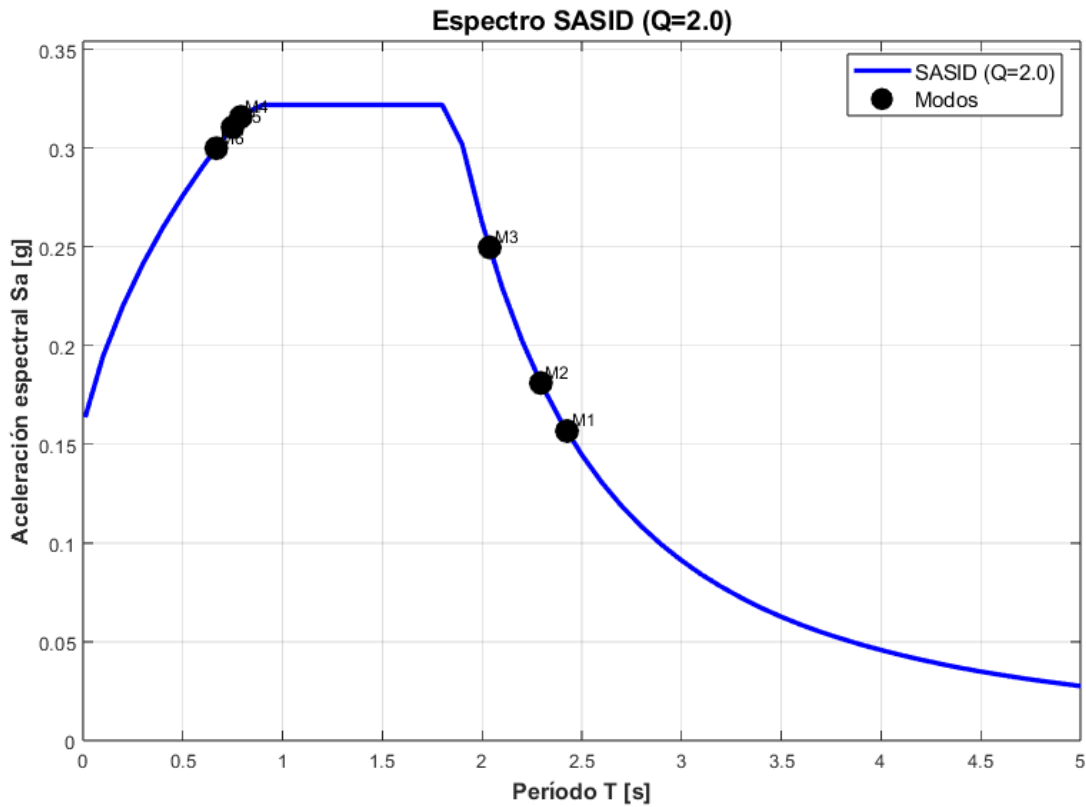


Figura 2.12: Espectro de aceleraciones obtenido del SASID para suelo Tipo III - Lago. Los puntos negros indican los primeros seis modos de vibrar del edificio.

ANÁLISIS ESPECTRAL - RESPUESTA MODAL

Para cada modo i seleccionado se interpola el espectro de diseño en el período modal T_i para obtener la aceleración espectral $S_a(T_i)$. El desplazamiento espectral es

$$S_{d,i} = S_a(T_i) \left(\frac{T_i}{2\pi} \right)^2. \quad (2.19)$$

La coordenada modal máxima se obtiene con

$$q_{i,\max} = \Gamma_i S_{d,i}. \quad (2.20)$$

Los desplazamientos nodales asociados al modo i son

$$\mathbf{u}_{i,\max} = \hat{\varphi}_i q_{i,\max}, \quad (2.21)$$

donde $\mathbf{u}_{i,\max}$ es el vector de desplazamientos máximos de los grados de libertad activos debidos únicamente al modo i . Las fuerzas modales se calculan como

$$\mathbf{F}_{i,\max} = \mathbf{K}_{\text{red}} \mathbf{u}_{i,\max} = \mathbf{M}_{\text{red}} \hat{\varphi}_i \Gamma_i S_a(T_i), \quad (2.22)$$

y representan las fuerzas inerciales máximas asociadas al modo correspondiente. Este proceso se repite para todos los modos con participación modal significativa, generando matrices de desplazamientos y fuerzas espectrales donde cada columna corresponde a un modo.

ANÁLISIS ESPECTRAL - COMBINACIÓN MODAL

Los desplazamientos máximos de cada modo se combinan mediante la regla de la raíz cuadrada de la suma de los cuadrados (SRSS) (Avilés López et al., 2016). Para el grado de libertad j , el desplazamiento combinado es

$$u_{j,\max} = \sqrt{\sum_i u_{ij,\max}^2}, \quad (2.23)$$

donde la suma se extiende sobre todos los modos considerados. La combinación asume que los máximos modales no ocurren simultáneamente, hipótesis válida cuando la separación entre períodos consecutivos es mayor al 10 % (Avilés López et al., 2016). Para estructuras con modos muy próximos, las NTC de Diseño por Sismo recomiendan emplear la Combinación Cuadrática Completa (CQC) (Gobierno de la Ciudad de México, 2017), que toma en cuenta la correlación entre modos mediante coeficientes que dependen de las relaciones de períodos y amortiguamiento. El método SRSS constituye el enfoque estándar para edificios de geometría regular con modos de vibrar suficientemente separados (Avilés López et al., 2016).

En el modelo implementado, los períodos de los primeros cinco modos oscilan entre 0.80 y 0.97 s. La diferencia entre modos consecutivos es inferior al 10 % en varios casos, condición que conforme a la NTC-Sismo 2020 exigiría el uso de CQC. Sin embargo, el código implementa únicamente SRSS. Esta decisión constituye una simplificación del modelo y debe considerarse al interpretar los resultados espectrales: la combinación SRSS puede subestimar la respuesta cuando los modos están correlacionados. La implementación de CQC queda fuera del alcance de este trabajo.

La combinación se aplica tanto a desplazamientos como a fuerzas internas, produciendo los vectores \mathbf{u}_{\max} y \mathbf{F}_{\max} que representan las demandas máximas esperadas durante el evento sísmico analizado.

ANÁLISIS ESPECTRAL - RESULTADOS ESPECTRALES

Los desplazamientos espectrales combinados se presentan organizados por nivel para facilitar su interpretación. Se calcula el desplazamiento promedio por piso en cada dirección principal, lo que permite identificar el desplazamiento máximo y el nivel donde ocurre. A continuación se genera una matriz de desplazamientos nodales que expande el valor reducido mediante las condiciones de diafragma y recupera los grados de libertad dependientes, proceso análogo al llevado a cabo en el análisis estático. Los desplazamientos relativos de entrepiso también se calculan. El script incorpora la opción de amplificación visual de los desplazamientos para su revisión detallada.

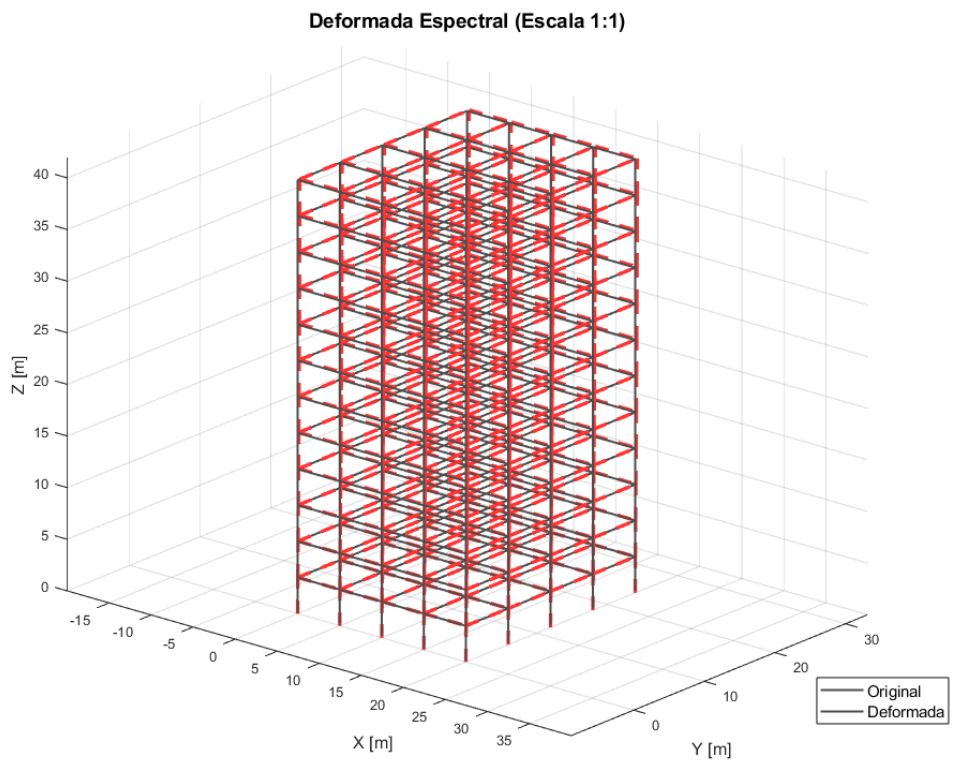


Figura 2.13: Deformada espectral bajo excitación en un eje. La estructura original se representa en negro y la deformada en rojo punteado.

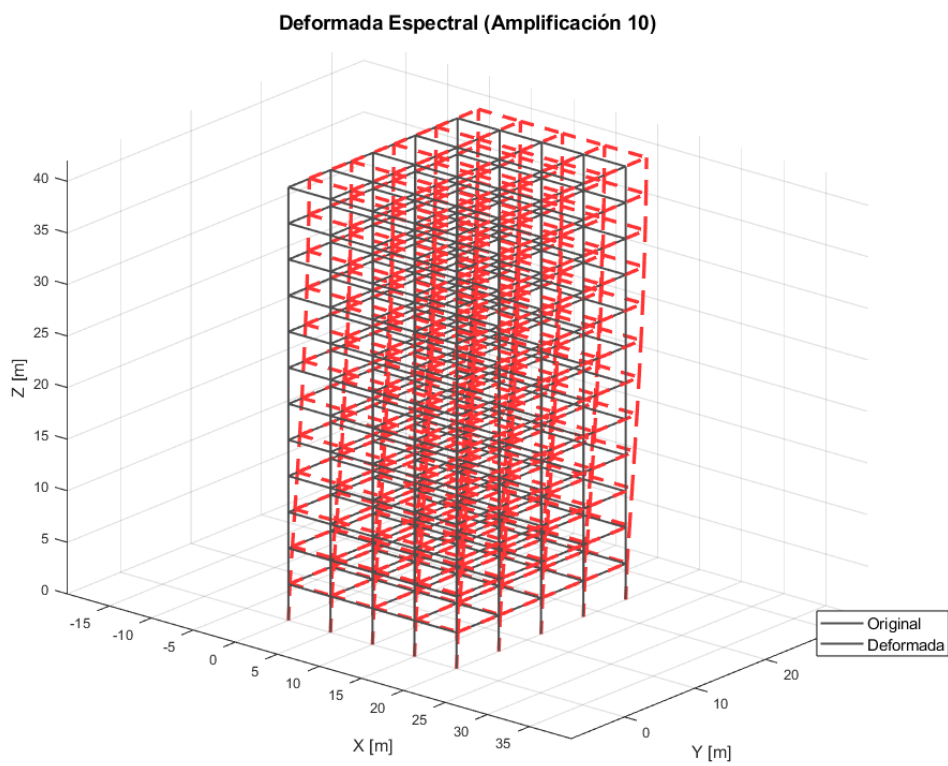


Figura 2.14: Deformada espectral bajo excitación en un eje con un factor de amplificación de $10\times$. La estructura original se representa en negro y la deformada en rojo punteado.

ANÁLISIS TIEMPO HISTORIA - CONFIGURACIÓN

En esta sección se establecen los parámetros numéricos para la integración temporal de las ecuaciones de movimiento. El coeficiente de amortiguamiento crítico es $\xi = 5\%$ para todos los modos de vibrar, valor recomendado por las NTC de Diseño por Sismo para estructuras de concreto reforzado sin dispositivos especiales de disipación de energía (Gobierno de la Ciudad de México, 2017). La integración temporal se realiza en coordenadas modales desacopladas, lo que supone que el amortiguamiento es clásico, igual para todos los modos e independiente del acoplamiento entre ellos. Esta es la hipótesis estándar en análisis modal con amortiguamiento proporcional (Chopra, 2007), distinta del amortiguamiento de Rayleigh utilizado en el análisis ISE, donde los coeficientes se calibran para dos modos de referencia. El acelerograma se aplica en una sola dirección a la vez; en este análisis se empleó la dirección Y como dirección de excitación principal, sin excitación simultánea en X .

El script utiliza los primeros 12 modos de vibrar calculados en el análisis modal previo, configuración que garantiza la inclusión de la masa participante significativa del sistema estructural.

ANÁLISIS TIEMPO HISTORIA - ACELEROGRAMA

El script permite trabajar con dos tipos de acelerogramas: sintético o de sismos reales. Para este trabajo se utilizó el acelerograma registrado por la SCT 2, correspondiente al sismo del 19 de septiembre de 2017. Esta estación se encuentra en la Zona III de la Ciudad de México y su registro es representativo de las condiciones de suelo blando, donde se presentan amplificaciones dinámicas por resonancia suelo-estructura. El archivo de datos contiene una columna de aceleraciones con paso de tiempo constante ($dt = 0.01$ s) y captura la fase inicial del sismo, el movimiento fuerte y el decaimiento gradual, características de los sismos intraplaca. Las aceleraciones están en cm/s^2 (Gal), unidad estándar de los registros de la red acelerográfica del CENAPRED. Antes de la integración, los valores se convierten a m/s^2 mediante el factor $1/100$, de modo que la aceleración máxima queda como $a_{g,\max} = 0.917 \text{ m/s}^2$ ($0.094g$), valor consistente con los registros de la estación SCT2 para el sismo del 19 de septiembre de 2017 (Instituto de Geofísica, UNAM, 2017). El script puede utilizar acelerogramas sintéticos en caso de estudios paramétricos o verificaciones numéricas adicionales.

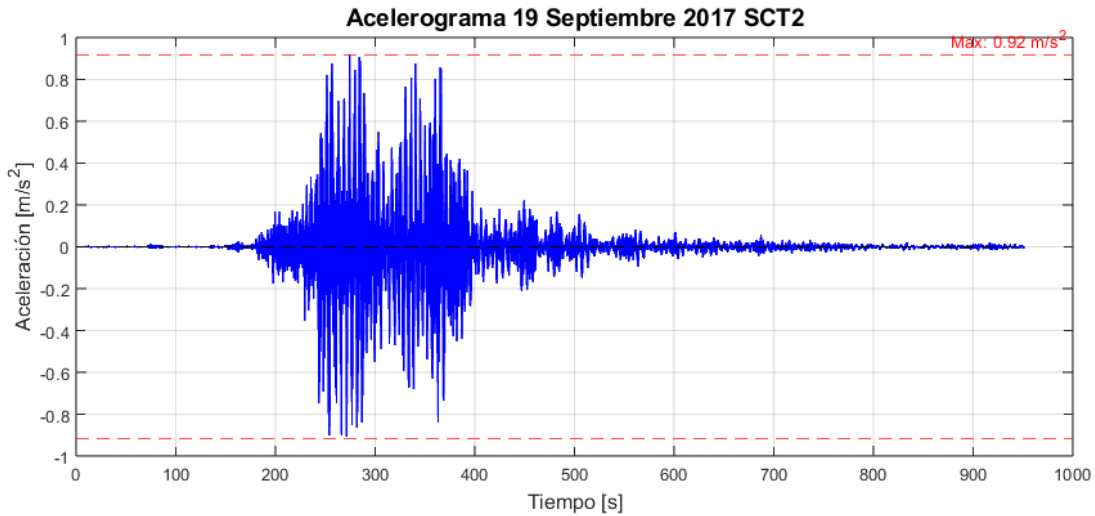


Figura 2.15: Acelerograma del sismo del 19 de septiembre del 2017. Registrado en la estación SCT2 (Instituto de Geofísica, UNAM, 2017).

ANÁLISIS TIEMPO HISTORIA - INTEGRACIÓN MODAL

La respuesta dinámica se calcula mediante el método de la superposición modal (Avilés López et al., 2016; Hori, 2011). Con este método, cada modo se resuelve como un oscilador de un grado de libertad independiente. Para el modo i , la ecuación diferencial ordinaria es

$$\ddot{q}_i + 2\xi\omega_i\dot{q}_i + \omega_i^2q_i = -\Gamma_i a_g(t), \quad (2.24)$$

donde $q_i(t)$ es la coordenada modal, ξ es el amortiguamiento modal constante e igual para todos los modos, ω_i es la frecuencia angular del modo, Γ_i es el factor de participación y $a_g(t)$ es la aceleración del terreno.

La integración numérica se lleva a cabo mediante el método de Newmark- β con parámetros $\gamma = 0.5$ y $\beta = 0.25$, conocido como esquema de aceleración promedio constante. Esta configuración garantiza estabilidad incondicional y precisión de segundo orden para sistemas lineales.

El script avanza para cada paso de tiempo desde $t = 0$ hasta el final del registro, con el paso de tiempo del acelerograma $dt = 0.01$ s. En cada incremento temporal se calculan las coordenadas modales $q_i(t)$, velocidades $\dot{q}_i(t)$ y aceleraciones $\ddot{q}_i(t)$ para todos los modos. La matriz de amortiguamiento modal se construye como una matriz diagonal con términos $2\xi\omega_i$, lo que representa amortiguamiento modal constante e igual

para todos los modos, independiente de la frecuencia, en coherencia con la hipótesis de amortiguamiento clásico adoptada en este análisis.

ANÁLISIS TIEMPO HISTORIA - RECONSTRUCCIÓN

Los desplazamientos totales de todos los grados de libertad activos se obtienen como suma de las contribuciones modales para cada instante de tiempo:

$$\mathbf{u}(t) = \sum_i \hat{\varphi}_i q_i(t). \quad (2.25)$$

Esta suma se realiza sobre todos los modos incluidos en el análisis dinámico. El resultado es una matriz \mathbf{U}_{hist} que contiene la historia temporal completa de los desplazamientos. De esta matriz se extraen los valores máximos absolutos para cada grado de libertad a lo largo de la duración del sismo:

$$u_{j,\text{max}} = \max_t |u_j(t)|. \quad (2.26)$$

El resultado es el vector $\mathbf{U}_{\text{hist,max}}$, que representa la envolvente de la respuesta estructural (Khosravikia et al., 2018). Los desplazamientos máximos obtenidos pueden compararse con los resultados del análisis espectral para verificar la consistencia entre ambos métodos. La diferencia entre ambos análisis suele ser inferior al 10 % para modos representados en el espectro de diseño, aunque puede ser mayor para modos con períodos fuera del rango dominante del acelerograma empleado (J. M. Mayoral y Pérez, 2024).

Como resultado se genera una gráfica que muestra los desplazamientos máximos y los instantes en que ocurren, para cada componente direccional (Figura 2.16).

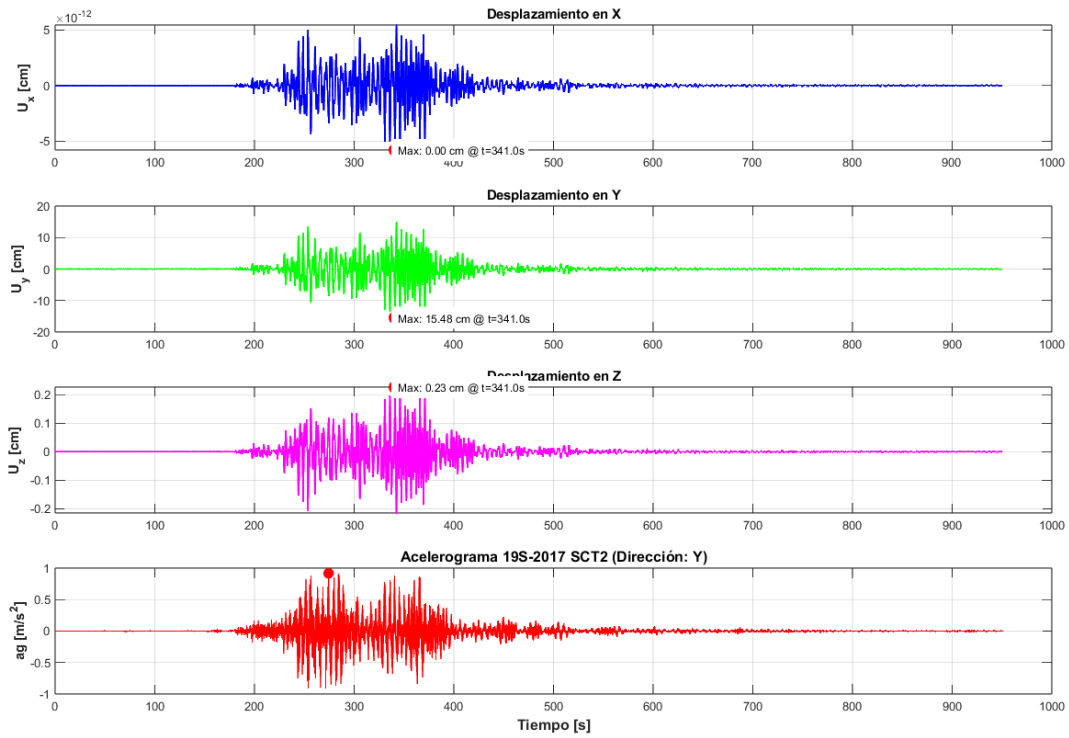


Figura 2.16: Historia de desplazamientos tiempo-historia en la estructura, provocados por el sismo del 19 de septiembre del 2017.

VISUALIZACIÓN ANIMADA

El script genera una animación tridimensional de la respuesta dinámica de la estructura durante el evento sísmico. La animación en formato de vídeo muestra la deformación a lo largo de los instantes del acelerograma. La estructura se grafica en color azul.

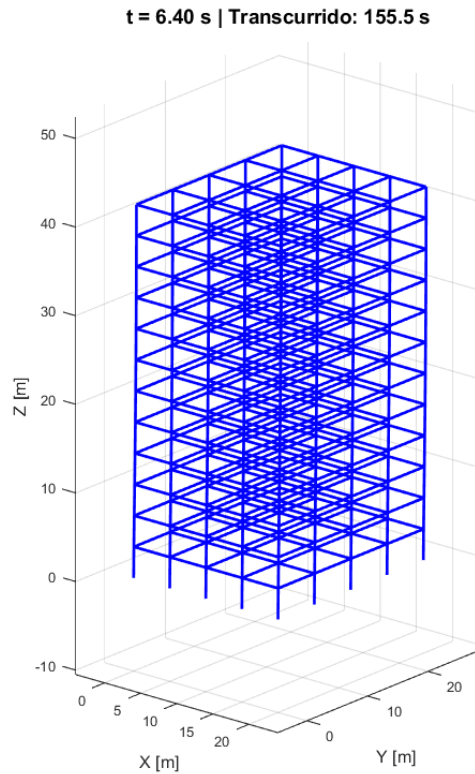


Figura 2.17: Fotograma representativo de la animación tridimensional generada por el código durante el análisis de historia de tiempo.

SALIDAS DEL ANÁLISIS DINÁMICO

El script genera los siguientes resultados según cada tipo de análisis, presentados de la manera descrita a continuación.

Para el análisis modal:

- Frecuencias naturales y períodos de vibración
- Formas modales normalizadas
- Factores de participación modal por dirección
- Masas modales efectivas y porcentajes de participación
- Verificación del 90 % de masa participante

Para el análisis espectral:

- Desplazamientos espectrales máximos por nivel y dirección
- Distorsiones de entrepiso
- Deformada espectral tridimensional
- Comparación con límites normativos

Para el análisis tiempo-historia:

- Historias temporales de desplazamiento
- Envolvente de desplazamientos máximos
- Instantes críticos de máxima demanda
- Comparación espectral vs. tiempo-historia
- Animación dinámica de la respuesta

El formato de salida del código es el siguiente:

```
>> analisis_dinamico
```

```
ANÁLISIS DINÁMICO MODAL
```

```
=====
```

```
Nodos
```

```
GDL
```

```
Matriz de masa:
```

```
Tamaño
```

Masa total

Grados de libertad:

Total

Restringidos

Libres

Matrices reducidas:

K_red

M_red

Resolviendo problema de autovalores...

Modos calculados

FRECUENCIAS Y PERÍODOS NATURALES

=====

Modo	Frecuencia	Período
------	------------	---------

Modos listados

Normalizando modos respecto a masa...

Modos normalizados

Verificando ortogonalidad de modos...

Error ortogonalidad masa (aceptable)

Error ortogonalidad rigidez (aceptable)

Calculando factores de participación...

Factores calculados

MASAS MODALES PARTICIPANTES

=====

Modo	X	Y	Z
------	---	---	---

Modos listados

Suma acumulada

=====

Generando espectro de diseño (SASID)...

Espectro cargado

Número de puntos

Rango de períodos

Aceleración espectral máxima

Factor Q base

Factor Q adicional

Factor Q efectivo

Valores espectrales en modos principales:

Modos evaluados

Calculando desplazamientos espectrales...

Desplazamientos modales calculados

Aplicando combinación SRSS...

Combinación SRSS aplicada

Desplazamientos espectrales por nivel:

```
=====
```

Nivel	Altura	Ux (cm)	Uy (cm)
0	0.00	x.xxxx	x.xxxx
...
12	42.00	x.xxxx	x.xxxx

```
=====
```

Generando gráfica del espectro...

Gráfica generada

Generando visualización de deformada...

Factor de amplificación

Deformación máxima

Visualización interactiva generada

Configurando análisis tiempo-historia...

Amortiguamiento crítico

Modos considerados

Cargando acelerograma...

Tipo de acelerograma

Conversión de unidades

Duración

Paso de tiempo

Número de puntos

Aceleración máxima

Generando gráfica del acelerograma...

Gráfica generada

Iniciando integración temporal (Newmark-beta)...

Parámetros Newmark

Dirección de excitación

Integrando modos

Integración completada

Reconstruyendo desplazamientos físicos...

Desplazamientos reconstruidos

Tamaño de la historia temporal

Desplazamientos máximos tiempo-historia por nivel:

```
=====
```

Nivel	Altura	Ux_max (cm)	Uy_max (cm)
0	0.00	x.xxxx	x.xxxx
...
12	42.00	x.xxxx	x.xxxx

```
=====
```

Desplazamientos máximos globales:

Ux

Uy

Uz

Reconstruyendo historia completa...

Historia completa reconstruida

Generando gráficas de respuesta temporal...

Nodo seleccionado

Gráficas generadas

Preparando animación 3D...

Duración

Factor de amplificación

Salto de frames

Frames totales

Ejecutando animación...

Animación completada

Tiempo simulado

Tiempo real

Frames renderizados

>>

2.2.3. ANÁLISIS DE INTERACCIÓN SUELO ESTRUCTURA ESTÁTICO-DINÁMICO

El Análisis de Interacción Suelo-Estructura está basado en el método de superposición de tres pasos propuesto en Avilés López et al. El script descompone el sistema acoplado suelo-estructura en subsistemas independientes y se estructura en cinco etapas fundamentales: (1) verificación del modelo estructural con base rígida, (2) definición de parámetros geotécnicos según la zonificación sísmica analizada, (3) cálculo de impedancias dinámicas del suelo, (4) evaluación del efecto de interacción suelo-estructura en los períodos naturales del edificio y, (5) integración temporal con el acelerograma del sismo del 19 de septiembre de 2017.

La estructura del código se explica en los diagramas siguientes.

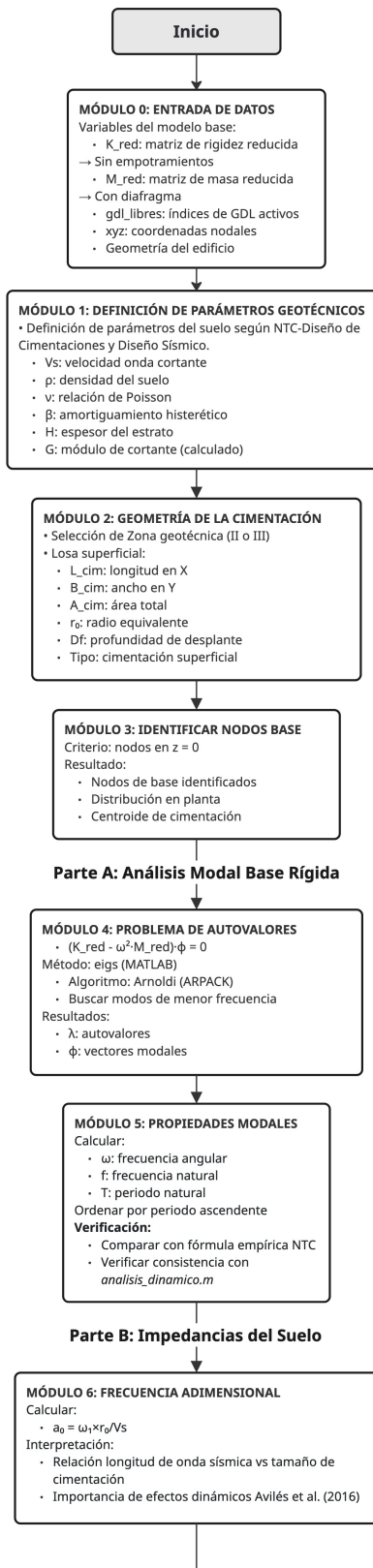


Figura 2.18: Diagrama de proceso del análisis de Interacción Suelo-Estructura. Continúa en la siguiente página.

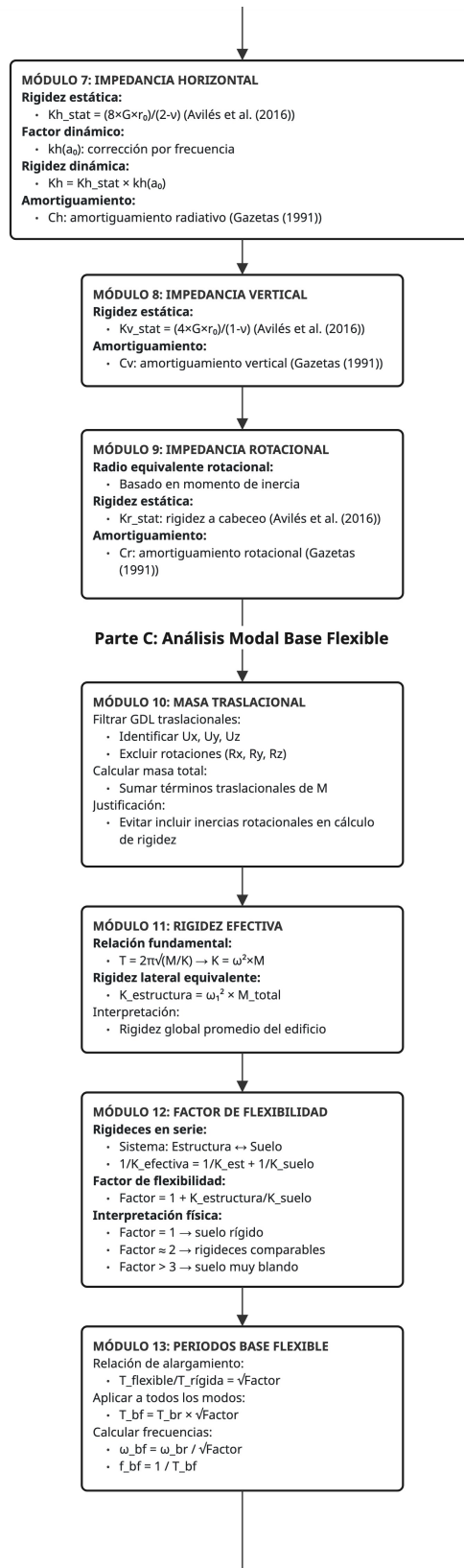


Figura 2.19: Continuación del diagrama de proceso del análisis de Interacción Suelo-Estructura. Continúa en la siguiente página.

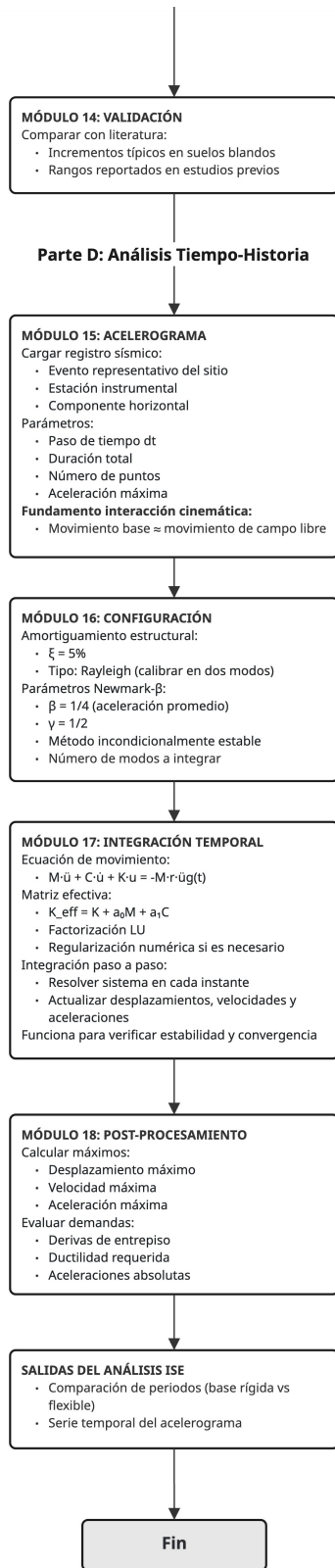


Figura 2.20: Continuación del diagrama de proceso del análisis de Interacción Suelo-Estructura.

DEFINICIÓN DE PARÁMETROS GEOTÉCNICOS

El código admite el uso de la Zona III - Lago o la Zona II - Transición, con sus respectivos parámetros. Las variables empleadas se obtienen de las NTC-Cimentaciones (2017), que especifican que, en ausencia de estudios locales de mecánica de suelos suficientes, puede adoptarse un valor de $V_s = 80$ m/s para la Zona III. La velocidad de onda cortante es un parámetro de suma importancia para el análisis, ya que se utiliza para calcular la rigidez al corte del suelo y la frecuencia adimensional que gobierna el comportamiento dinámico de las impedancias. La densidad del suelo también se obtuvo de las NTC-Cimentaciones (2017). La arcilla lacustre saturada de la Zona III se comporta como material cuasi-incompresible en condiciones no drenadas, lo que justifica físicamente $\nu \approx 0.5$. La elección de $\nu = 0.499$ refleja esta propiedad del material y evita simultáneamente las indeterminaciones numéricas asociadas a la incompresibilidad exacta en las expresiones de impedancia horizontal.

El amortiguamiento histérico de $\beta = 0.03$ se adoptó siguiendo las NTC-Cimentaciones (2017). El módulo de cortante (Ec. 2.27) se obtuvo mediante la relación fundamental de ondas elásticas (Kramer, 1996):

$$G = \rho V_s^2 \quad (2.27)$$

El espesor del estrato H es de 50 m, como valor representativo promedio de la Zona III (Cruz-Atienza et al., 2016; Efraín Ovando-Shelley y Santoyo, 2013; J. M. Mayoral y Pérez, 2024). Todos los parámetros son modificables para análisis paramétrico dependiendo de la información disponible.

A continuación se presenta la tabla resumen de los parámetros geotécnicos utilizados.

Cuadro 2.1: Parámetros geotécnicos utilizados.

Parámetro	Símbolo	Valor	Unidad	Fuente
Velocidad de onda cortante	V_s	80	m/s	NTC-Cimentaciones (2017)
Densidad del suelo	ρ	1,250	kg/m ³	NTC-Cimentaciones (2017)
Relación de Poisson	ν	0.499	-	NTC-Cimentaciones (2017)
Amortiguamiento histérico	β	0.03	-	NTC (2017) y Avilés López et al., 2016
Espesor del estrato	H	50	m	NTC-Cimentaciones (2017)
Módulo de cortante	G	8,000	kPa	Calculado

GEOMETRÍA DE LA CIMENTACIÓN

En esta sección del código se introducen las dimensiones de la losa: su longitud en el eje X , en el eje Y y la profundidad de desplante. La cimentación se idealiza como una losa superficial de planta rectangular. A partir de estas dimensiones se calcula el radio equivalente y se identifican los nodos de la estructura ubicados en la base, con coordenada $z = 0$. Avilés López et al. justifica el uso del radio equivalente para geometrías arbitrarias, con un error inferior al 15 %.

ANÁLISIS MODAL CON BASE RÍGIDA (MÓDULO 4-5)

Previo al cálculo de los efectos de ISE, se establece como referencia el análisis modal de la estructura con empotramiento perfecto en la base. El problema de autovalores generalizado se resuelve en MATLAB mediante la función `eigs`, que implementa el método de Arnoldi, optimizado para matrices dispersas. Se utiliza la opción `smallestabs` para buscar los modos de menor frecuencia, que corresponden a los períodos fundamentales de la estructura.

Los autovalores λ_i se convierten en propiedades modales: frecuencia angular, frecuencia natural y período natural. Con el fin de verificar los períodos obtenidos, se utiliza la fórmula empírica de las NTC-Sismo (2017):

$$T_a = 0.05 \times H^{3/4} \quad (2.28)$$

Los resultados obtenidos en este proceso deben ser idénticos a los del script `analisis_dinamico.m`, lo que permite verificar la consistencia del modelo.

CÁLCULO DE IMPEDANCIAS DINÁMICAS (MÓDULO 6-9)

Las impedancias dinámicas dependen de la frecuencia y representan la resistencia elástica del suelo y su capacidad de disipar energía por radiación de ondas. Cuantifican la relación entre las fuerzas y momentos

aplicados a la cimentación y los desplazamientos y rotaciones resultantes.

La frecuencia adimensional representa la relación entre la longitud de onda sísmica y la dimensión característica de la cimentación. El script calcula a_0 a partir de la frecuencia fundamental de la estructura para evaluar si los efectos dinámicos tienen impacto significativo.

El amortiguamiento por radiación en el modo traslacional horizontal se calcula mediante la siguiente expresión, donde $a_0 = \tilde{\omega}_e r_0 / V_s$ es la frecuencia adimensional evaluada en la frecuencia efectiva convergida del sistema, y $c_b = 0.6$ es el coeficiente adimensional de Gazetas para traslación horizontal de cimentación circular superficial:

$$C_b = K_{b,\text{stat}} \cdot c_b \cdot a_0 \cdot \frac{r_0}{V_s} \quad (2.29)$$

La rigidez estática horizontal se obtiene mediante la Ec. 2.30, derivada de la solución de Mindlin-Reissner para carga tangencial horizontal sobre un semiespacio elástico, extendida al caso del disco circular rígido por Luco y Westmann (1971) y sistematizada por Gazetas. Esta expresión forma parte del conjunto de rigideces estáticas de referencia adoptado por la NTC-Sismo 2020 y por Avilés López et al.

Para corregir la rigidez a frecuencias distintas de cero se utiliza el factor dinámico $k_b(a_0)$.

$$K_{b,\text{stat}} = \frac{8 G r_0}{2 - \nu} \quad (2.30)$$

La impedancia vertical (rigidez estática) se calcula con la siguiente expresión:

$$K_{v,\text{stat}} = \frac{4 G r_0}{1 - \nu} \quad (2.31)$$

La impedancia rotacional o de cabeceo requiere el radio equivalente basado en el momento de inercia de la cimentación (Ec. 2.32). La rigidez estática correspondiente se calcula mediante la Ec. 2.33, y para el amortiguamiento se utiliza $c_r = 0.4$ (Gazetas, 1991):

$$r_{0r} = \left(\frac{I_x}{A} \right)^{1/4} \quad (2.32)$$

$$K_{r,\text{stat}} = \frac{8 G r_0^3}{3(1 - \nu)} \quad (2.33)$$

ANÁLISIS MODAL CON BASE FLEXIBLE (MÓDULO 10-13)

En un análisis ISE completo la estructura con base flexible se resuelve en el dominio de la frecuencia, donde las impedancias del suelo, función de ω , se incorporan directamente a las matrices K y C (Avilés López et al., 2016):

$$[K + i\omega C - \omega^2 M] \{u(\omega)\} = \{F(\omega)\} \quad (2.34)$$

En este trabajo se implementa una aproximación energética simplificada que permite estimar el efecto ISE en los períodos naturales sin resolver el sistema en el dominio de la frecuencia, lo que facilita el uso de espectros de respuesta estándar de campo libre (Avilés López et al., 2016, NTC-Sismo 2017).

El script estima la masa efectiva del sistema identificando mediante un filtro los grados de libertad traslacionales (U_x, U_y, U_z) y excluyendo las rotaciones nodales (R_x, R_y, R_z). Este paso es necesario porque la matriz de masa contiene tanto términos de masa traslacional como términos de inercia rotacional nodal; incluirlos sin filtrar alteraría los resultados de las operaciones posteriores. Cabe notar que el grado de libertad de rotación del diafragma alrededor del eje vertical, que concentra la inercia rotacional del piso, se trata de forma separada a través de la condensación por diafragma rígido y no se ve afectado por este filtrado.

A partir del período T_1 y la masa filtrada, se estima la rigidez lateral equivalente del edificio mediante la Ec. 2.35:

$$K = \omega^2 M = \left(\frac{2\pi}{T} \right)^2 M \quad (2.35)$$

Esta rigidez representa la resistencia global del edificio al desplazamiento lateral, condensada en un único valor que captura el comportamiento del modo fundamental.

El período efectivo del sistema con base flexible se determina mediante la expresión de suma cuadrática (NTC-Sismo 2020):

$$\tilde{T}_e^2 = T_e^2 + T_b^2 + T_r^2 \quad (2.36)$$

donde $T_b = 2\pi\sqrt{W_e/(gK_b)}$ representa el período que tendría la estructura si su base solo pudiese trasladarse, y $T_r = 2\pi\sqrt{W_eH_e^2/(gK_r)}$ el período correspondiente al modo de cabeceo puro. W_e es el peso efectivo modal, H_e la altura efectiva, y K_b, K_r las rigideces de traslación y rotación de la cimentación evaluadas en la frecuencia $\tilde{\omega}_e = 2\pi/\tilde{T}_e$ mediante iteración. Esta expresión captura simultáneamente la flexibilidad traslacional y rotacional del suelo; para estructuras esbeltas sobre suelo blando, la contribución del cabeceo suele ser dominante en la elongación del período Avilés López et al., 2016. Los valores de K_b y K_r se obtienen conforme a la Tabla 8.2.1 de la NTC-Sismo 2020.

El factor de flexibilidad se define como

$$\text{Factor} = 1 + \frac{K_{\text{estructura}}}{K_{\text{suelo}}} \quad (2.37)$$

y cuantifica el grado en que el sistema se vuelve más flexible por la deformabilidad del suelo. Un factor de 1.0 indica suelo infinitamente rígido (sin ISE), un factor de 2.0 indica ISE moderado y un factor mayor a 3.0 indica ISE significativo (Avilés López et al., 2016; Khosravikia et al., 2018).

El alargamiento del período se calcula como

$$\frac{\tilde{T}_e}{T_e} = \sqrt{\text{Factor}} \quad (2.38)$$

La Ec. 2.38 es consecuencia directa de la definición del período en función de la rigidez efectiva del sistema acoplado y se aplica al modo fundamental, que concentra la mayor participación de masa en la dirección de

análisis. Para modos superiores, el efecto ISE puede diferir porque las impedancias dinámicas dependen de la frecuencia de excitación (Avilés López et al., 2016).

VALIDACIÓN CON LITERATURA (MÓDULO 14)

El incremento del período obtenido por la ISE se compara con valores reportados en la literatura para casos similares en la Zona III - Lago. Estos valores oscilan entre +20 % y +55 % de incremento del período (Avilés López et al., 2016; J. Mayoral et al., 2019; Quinde y Reinoso, 2020).

ANÁLISIS TIEMPO HISTORIA (MÓDULOS 15-18)

El acelerograma empleado es el sismo del 19 de septiembre de 2017 registrado en la estación SCT2. El amortiguamiento de la estructura se establece en $\xi = 5\%$ conforme a las NTC-Sismo (2020) para el diseño de estructuras de concreto en el rango de comportamiento elástico. A diferencia del análisis dinámico modal, donde se adopta amortiguamiento modal constante, en el análisis ISE el amortiguamiento se implementa mediante el modelo de Rayleigh (Ec. 2.39) (Hori, 2011):

$$\mathbf{C} = \alpha_R \mathbf{M} + \beta_R \mathbf{K} \quad (2.39)$$

Los coeficientes α_R y β_R se calibran especificando el mismo porcentaje de amortiguamiento crítico en dos modos de referencia, comúnmente los modos 1 y 3 (Avilés López et al., 2016; Hori, 2011). Esta distinción respecto al análisis dinámico modal es relevante: el amortiguamiento de Rayleigh introduce dependencia en la frecuencia, de modo que los modos superiores e inferiores a los de referencia tienen un amortiguamiento efectivo diferente de ξ .

Para la integración temporal se utiliza el método de Newmark- β con parámetros $\beta = 1/4$ y $\gamma = 1/2$, que corresponden al esquema de aceleración promedio constante (Hori, 2011; Oñate y Rojek, 2004). Este esquema es incondicionalmente estable y no introduce amortiguamiento numérico artificial, con orden de precisión de segundo orden.

El script resuelve la ecuación de movimiento (Ec. 2.40) paso a paso mediante la matriz efectiva (Ec. 2.41), que se factoriza una única vez al inicio mediante descomposición (Hori, 2011; Vicencio et al., 2023):

$$\mathbf{M} \ddot{u} + \mathbf{C} \dot{u} + \mathbf{K} u = -\mathbf{M} r \ddot{u}_g(t) \quad (2.40)$$

$$\mathbf{K}_{\text{eff}} = \mathbf{K} + a_0 \mathbf{M} + a_1 \mathbf{C} \quad (2.41)$$

En cada paso de tiempo se calcula la fuerza efectiva, se resuelve el sistema lineal y se actualizan los desplazamientos, velocidades y aceleraciones Hori, 2011. Se incluye una regularización numérica en la diagonal de \mathbf{K}_{eff} para evitar singularidades por condicionamiento deficiente (Avilés López et al., 2016; Celigüeta-Lizarza, 2011; McGuire et al., 2000). Al completar la integración, el script extrae los valores máximos de desplazamiento, velocidad y aceleración, que permiten evaluar los desplazamientos máximos de entrepiso.

VISUALIZACIÓN Y RESULTADOS (MÓDULOS 19 Y 20)

Como resultado del script se genera la gráfica de comparación de períodos (Figura 2.21), que muestra el efecto de Interacción Suelo-Estructura mediante barras agrupadas de base rígida vs. base flexible para los primeros modos de vibrar. La gráfica del acelerograma (Figura 2.15) muestra la historia del movimiento del terreno.

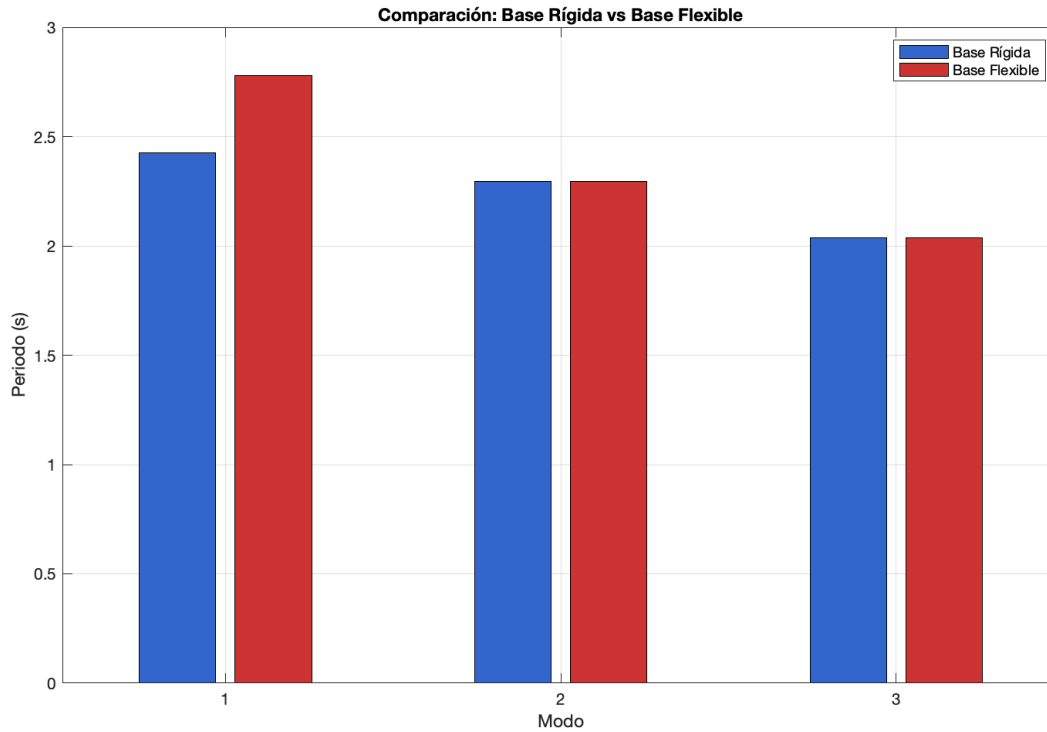


Figura 2.21: Comparación de períodos para mostrar efectos de Interacción Suelo-Estructura en los primeros modos de vibrar.

Finalmente, los resultados numéricos se presentan en forma de tabla comparativa con los períodos de base rígida, los períodos de base flexible, el incremento absoluto (ΔT) y el incremento porcentual ($\Delta T/T$) para los primeros modos de vibrar. A continuación se muestra el formato de salida utilizado, independientemente de los resultados obtenidos.

ANÁLISIS ISE - Método de Superposición

 PASO 0: Cargando modelo estructural...

Ejecutando scripts previos...

[modelo cargado]

- Nodos

- GDLs libres

PASO 1: Parámetros Zona III

$V_s, \rho, \nu, \beta_s, G$

Ref: NTC-Cimentaciones (2017)

PASO 2: Cimentación

Dimensiones losa: $L \times B$ m

Radio equivalente traslacional: r_0

Ref: NTC-Sismo 2020, ec. 8.2.13

PASO 3: Identificar nodos base

Nodos en $z=0$

PASO 4: Análisis modal BASE RÍGIDA

Primeros 3 modos:

Modo 1: T, f

Modo 2: T, f

Modo 3: T, f

PASO 5: Geometría de cimentación e impedancias estáticas

Ref: Gazetas (1991); Avilés López et al. (2016)

Formulación: disco circular rígido sobre semiespacio elástico

Radios equivalentes:

r0 (traslación): [= sqrt(A/pi)]
r0r (rotación X): [= (4Ix/pi)^(1/4)]
r0r (rotación Y): [= (4Iy/pi)^(1/4)]
r0r (promedio)

Rigideces estáticas:

$$K_h = 8 \cdot G \cdot r_0 / (2 - \nu)$$

$$K_r = 8 \cdot G \cdot r_{0r}^3 / [3 \cdot (1 - \nu)]$$

Coefficientes de amortiguamiento radiativo (evaluados en iteración):

ch (traslación horizontal, Gazetas 1991)

cr (cabeceo, Gazetas 1991)

Verificación amortiguamiento por radiación:

Frecuencia resonante del estrato: fs

Frecuencia fundamental estructura: f1

[OK/ADVERTENCIA según relación f1-fs]

PASO 6: Análisis modal BASE FLEXIBLE

MÉTODO: Periodo efectivo por suma de cuadrados (NTC-Sismo 2020)

Ref: NTC-Sismo 2020, ecs. 8.2.7-8.2.12

Avilés López et al. (2016), Sec. A4.3

Dirección de análisis: Y

Modo traslacional dominante en Y: modo N

Masa efectiva modal (M*): ton

Peso efectivo modal (W^*): kN

Altura efectiva (H_e): m

ITERACIÓN - Impedancias evaluadas en $\omega = 2\pi/T^*$

Iter	T^* (s)	T_h (s)	T_r (s)	Error
1	x.xxxx	x.xxxx	x.xxxx	x.xxe-x
2	x.xxxx	x.xxxx	x.xxxx	x.xxe-x
...				

[OK] Convergencia en N iteración(es)

$$T^{*2} = T_e^2 + T_h^2 + T_r^2 \quad (\text{solución iterada})$$

T_e (base rígida):

T_h (traslación):

T_r (cabeceo):

T^* (base flexible):

$$dT = +x.xxxx \text{ s} \quad (dT/T_e = +x.x\%)$$

Contribución al incremento $T^{*2} - T_e^2$:

Traslación (T_h^2): x.x%

Cabeceo (T_r^2): x.x%

Rango esperado Zona III: +20% a +55%

Refs: Avilés et al. (2016); Mayoral et al. (2019)

Amortiguamiento efectivo ($\omega \sim$ convergida):

- xi_e (estructura)
- xi~h (traslación)
- xi~r (cabeceo)
- xi~ (sistema)

Comparación de periodos (modo 1 modificado por ISE):

Modo	T_br (s)	T_bf (s)	dT (s)	dT/T	
1	x.xxxx	x.xxxx	+x.xxxx	+x.x%	<- ISE
2	x.xxxx	x.xxxx	0.000	0.0%	
3	x.xxxx	x.xxxx	0.000	0.0%	

NOTA: Los modos superiores no se modifican por ISE.

Ref: NTC-Sismo 2020, Sec. 8.2; Avilés López et al. (2016)

PASO 7: Acelerograma 19S-2017

Duración, aceleración máxima

Ref: CENAPRED, estación SCT2

PASO 8: Análisis tiempo-historia (base rígida)

Excitación: X e Y simultáneos (mismo acelerograma, en fase)

Regularización M_red: GDLs con masa=0, masa ficticia añadida

Amortiguamiento Rayleigh: alpha, beta
(Calibrado para xi=5% en modos 1 y 3)

GDLs en vector *iota_x* (Ux activos)

GDLs en vector *iota_y* (Uy activos)

GDLs restantes (rotaciones, Uz): no excitados

Factorizando *K_eff* (LU)...

Integrando pasos con Newmark-beta (aceleración promedio)...

10% 20% 30% 40% 50% 60% 70% 80% 90% 100% [OK]

u_max horizontal (envolvente X-Y): x.xx cm

PASO 9: Gráficas

Gráfica guardada: resultados_ise/periodos.png

ANÁLISIS COMPLETADO

RESUMEN ANÁLISIS ISE:

+-----+

Dirección de análisis:	Y
Modo traslacional dominante en Y:	modo N
Periodo fundamental (base rígida):	T1
Periodo fundamental (base flexible):	T~
Incremento total por ISE:	dT/T1
Contribución traslacional (Kh):	x.x%
Contribución rotacional (Kr / cabeceo):	x.x%

Desplazamiento máximo horizontal: u_{max}
+-----+
Rango típico Zona III (literatura): +20% a +55%
Refs: Avilés et al. 2016; Mayoral et al. 2019

PASO SIGUIENTE: Validar con SAP2000

3

Análisis de resultados

El error porcentual de cada cantidad comparada entre plataformas se calcula mediante:

$$\%e = \left| \frac{X_{\text{MATLAB}} - X_{\text{SAP2000}}}{X_{\text{SAP2000}}} \right| \times 100 \quad (3.1)$$

donde X_{MATLAB} y X_{SAP2000} representan el valor de la cantidad de interés obtenido por cada plataforma. Se comparó, como se había mencionado, contra los resultados obtenidos en SAP2000 v26.3.0.

Los resultados numéricos de cada análisis se presentan en las tablas y figuras de este capítulo. Las deformadas, formas modales y respuestas temporales obtenidas en ambas plataformas se muestran en las figuras correspondientes a cada caso de análisis, donde pueden verificarse cualitativamente la forma de la respuesta y la

consistencia entre modelos.

3.1. MODELO DE VALIDACIÓN EN SAP2000

El modelo de validación se construyó en SAP2000 v26.3.0 con la misma geometría, propiedades mecánicas y condiciones de frontera que el modelo implementado en MATLAB. La retícula estructural tiene 5 ejes en dirección X (A-E) y 5 ejes en dirección Y (1-5), con columnas de 50×50 cm y vigas de 50×30 cm en los 12 niveles de 3.5 m de altura. La Figura 3.1 muestra la vista tridimensional del modelo; en ella se distinguen las columnas (amarillo), las vigas (azul) y los empotramientos de la base (símbolos verdes), cuya disposición es idéntica a la del modelo MATLAB.

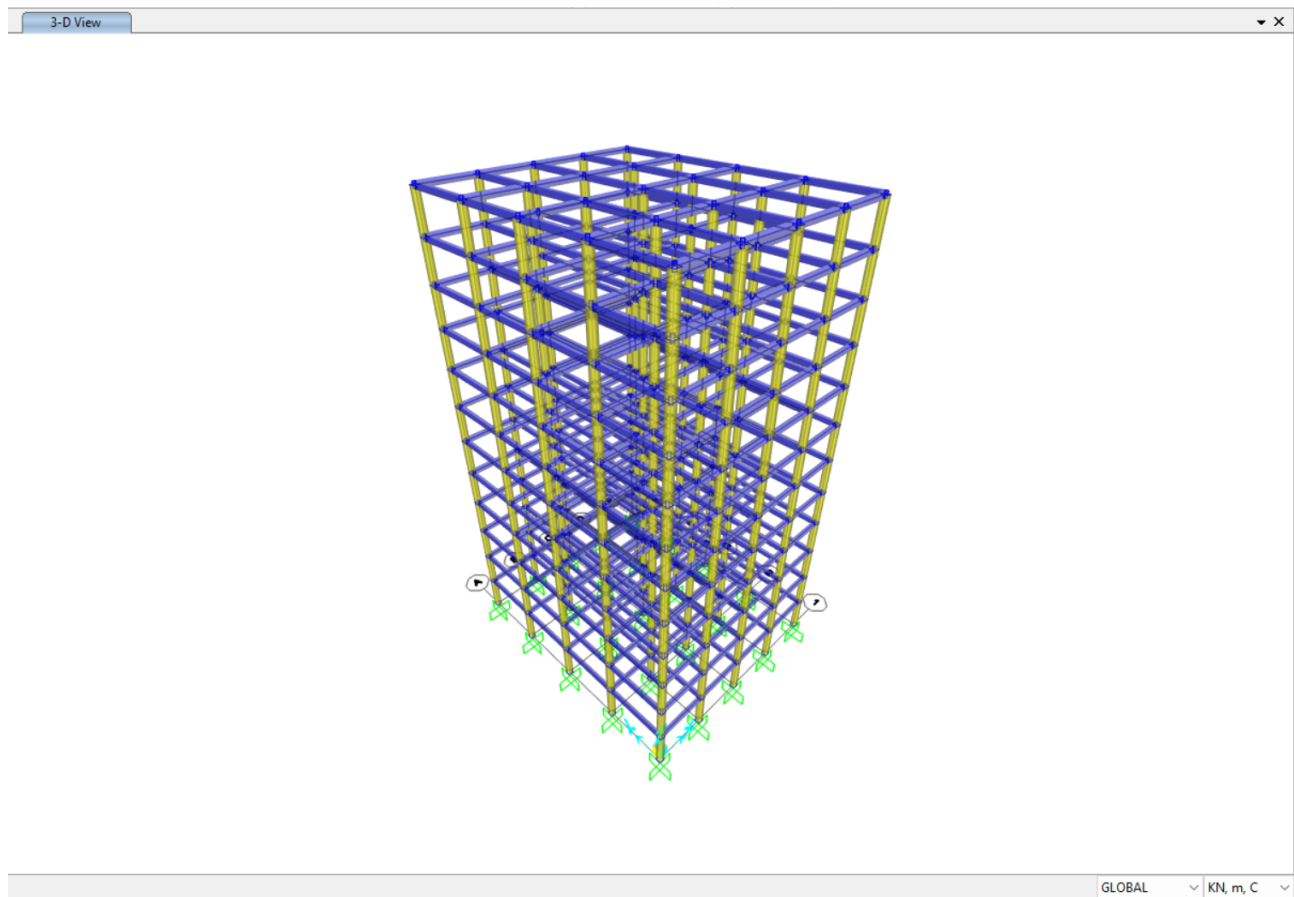


Figura 3.1: Vista tridimensional del modelo de base rígida en SAP2000. Las columnas se representan en amarillo y las vigas en azul. Los símbolos verdes en la base corresponden a los empotramientos que restringen los seis grados de libertad de cada nodo a nivel $z = 0$.

La planta del modelo a nivel de azotea ($Z = 42$ m) se presenta en la Figura 3.2. La retícula de cuatro crujeías en cada dirección —con claros de 5 m en X y 6 m en Y — produce una planta de 20×24 m, que corresponde

exactamente a las dimensiones de la losa de cimentación utilizadas en el cálculo de las impedancias del suelo.

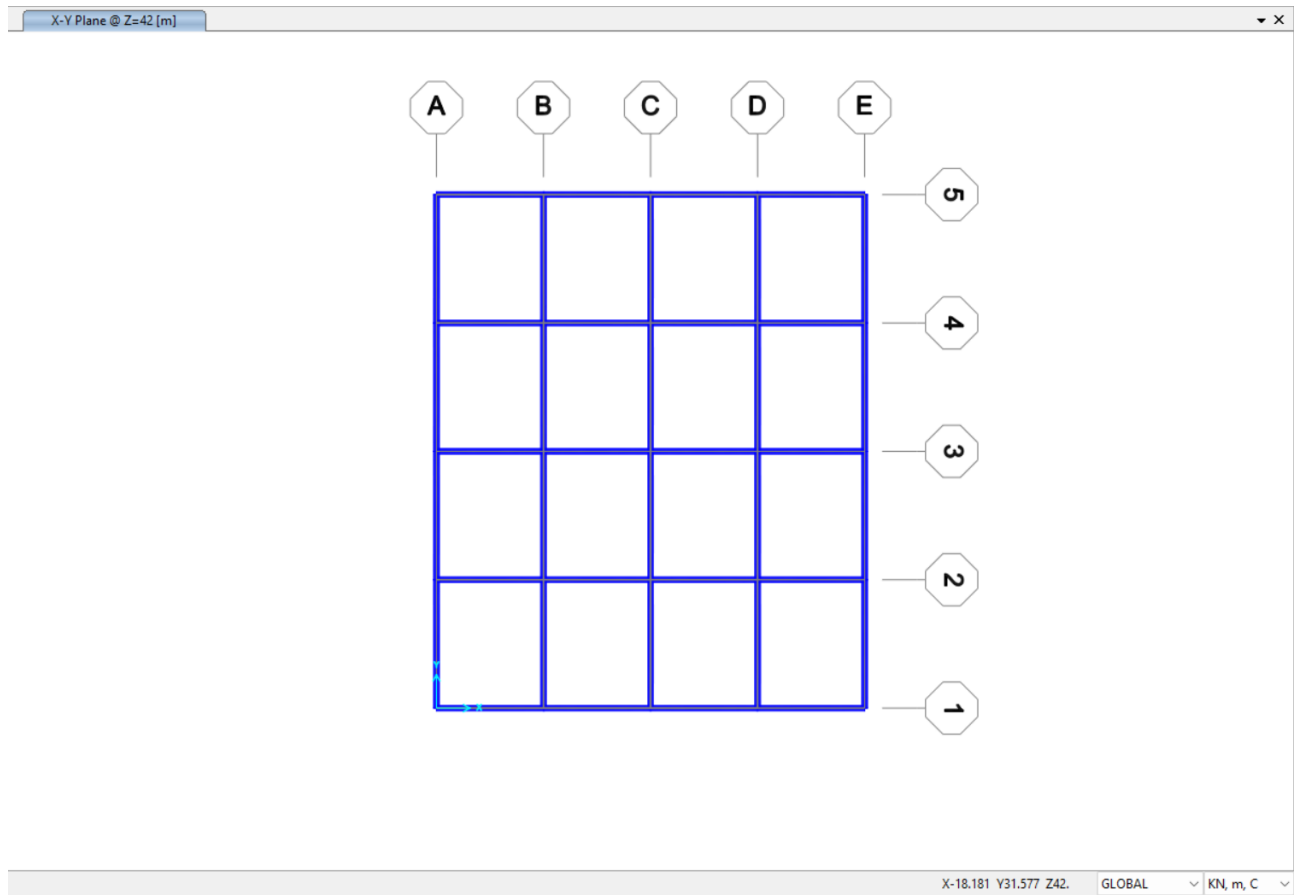


Figura 3.2: Planta del modelo en el plano X - Y a $Z = 42$ m (nivel de azotea). Los cinco ejes en X (A-E) y los cinco en Y (1-5) definen la retícula ortogonal con claros de 5 m y 6 m respectivamente.

La Figura ?? muestra el marco interior en el plano X - Z . La regularidad de la distribución de vigas y columnas a lo largo de los 12 niveles confirma que no existen discontinuidades geométricas en la elevación. La hipótesis de diafragma rígido por nivel se sustenta en que el sistema de entrepiso consiste en una losa maciza de concreto reforzado, cuya rigidez en el plano es significativamente mayor que la rigidez lateral de los marcos, condición reconocida en la literatura (Chopra2007) y admitida por la NTC-Sismo 2020 para este tipo de sistemas estructurales. Los apoyos de la base se ubican en todos los nodos a $z = 0$, sin excepción.

(NTC-Sismo 2020). Los apoyos de la base se ubican en todos los nodos a $z = 0$, sin excepción.

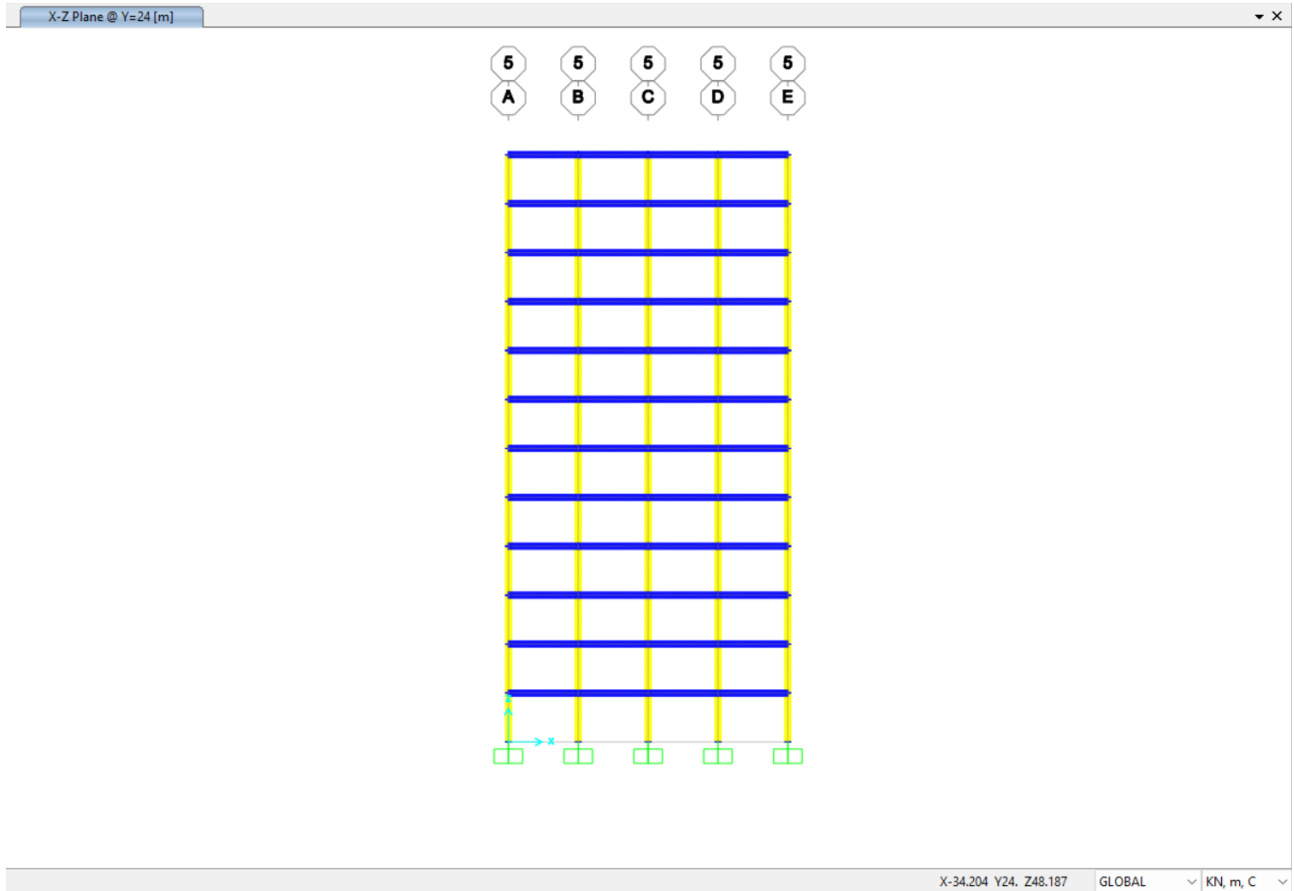


Figura 3.3: Marco en el plano $X-Z$ a $Y = 24$ m. La distribución regular de elementos en los 12 niveles de 3.5 m de altura confirma la ausencia de irregularidades en elevación. Los cuadros verdes en la base son los apoyos de empotramiento.

El detalle de la zona de cimentación (Figura 3.4) permite verificar que las dimensiones de las secciones transversales ingresadas en SAP2000 corresponden a las del modelo MATLAB: columnas cuadradas de 50 cm de lado y vigas rectangulares de 50×30 cm. Cada nodo de la base tiene asignado un empotramiento que impide desplazamientos y rotaciones en las tres direcciones.

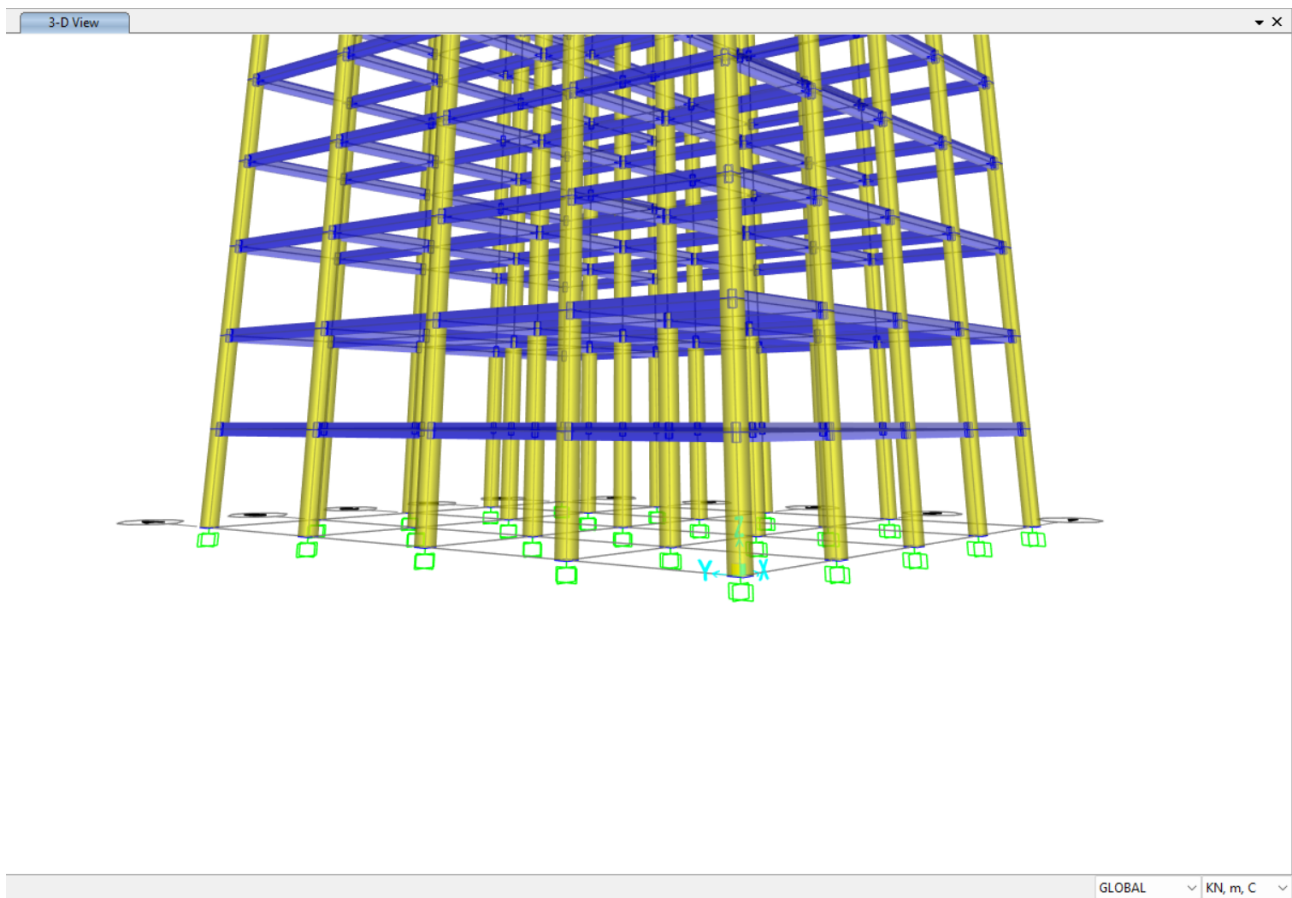


Figura 3.4: Detalle de la zona de cimentación del modelo en SAP2000 con renderizado de secciones. Las secciones de columnas (50×50 cm) y vigas (50×30 cm) son visibles en la representación tridimensional. Los cuadros verdes en cada nodo de la base corresponden a los empotramientos perfectos.

La masa de las losas se incorporó como carga distribuida equivalente de 12.84 kN/m sobre las vigas de cada nivel, activada como fuente de masa mediante *Specified Load Pattern* $\times 1.0$. Este valor resulta de dividir la carga superficial de 600 kg/m^2 entre la longitud tributaria de cada viga. El procedimiento y su justificación se describen en la Sección 3.4.1; la Figura 3.5 muestra que la carga es uniforme en todos los niveles, lo que es consistente con la hipótesis de carga muerta constante por entrepiso adoptada en MATLAB.

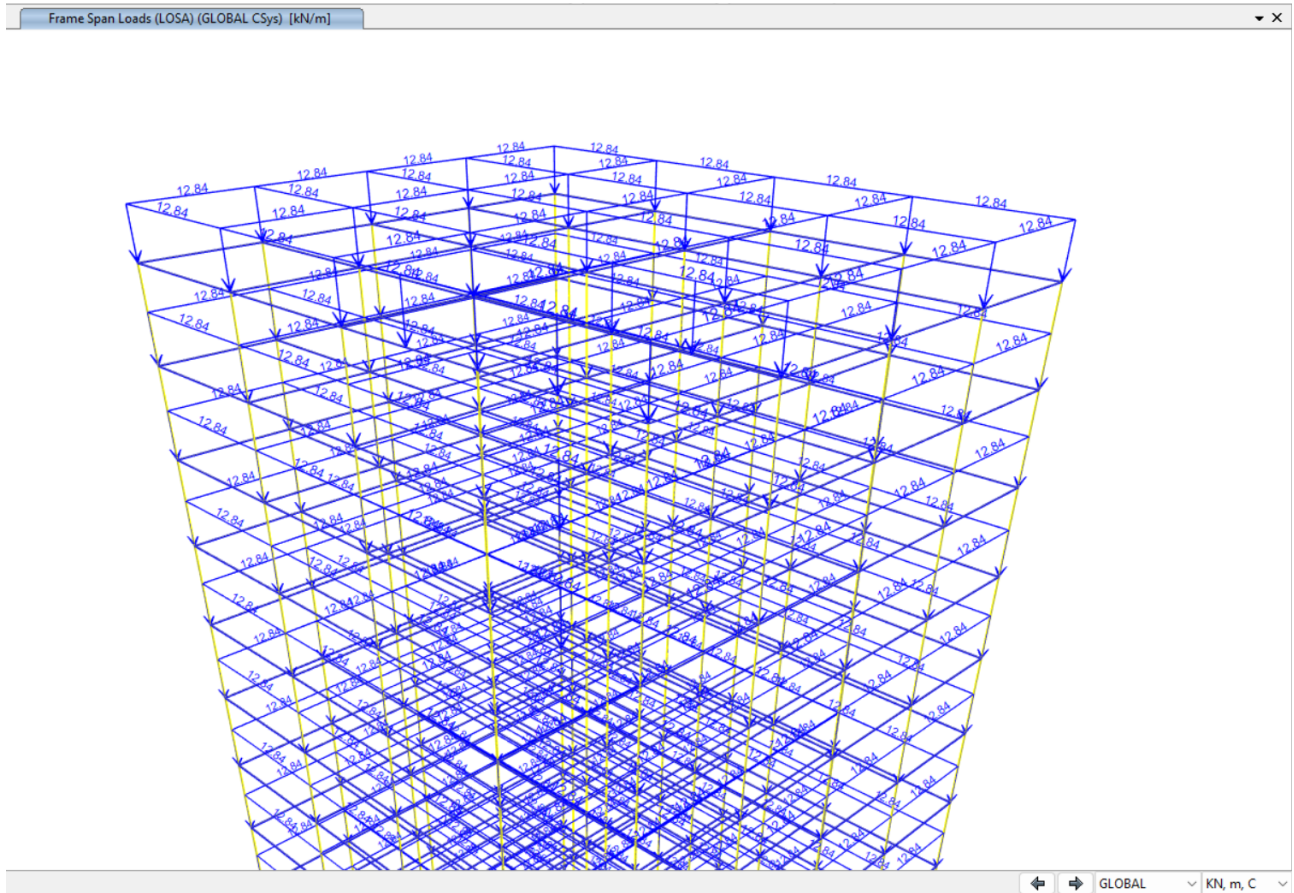


Figura 3.5: Cargas distribuidas de losa (caso LOSA) aplicadas sobre las vigas del modelo en SAP2000. La magnitud uniforme de 12.84 kN/m en todos los niveles es consistente con la carga superficial de 600 kg/m² distribuida según el área tributaria de cada viga.

3.2. COMPARACIÓN CUANTITATIVA DE LOS RESULTADOS OBTENIDOS

Con el fin de modelar adecuadamente, fueron establecidas condiciones de modelado equivalentes. Los supuestos comunes para cada caso, también mencionados anteriormente, son: comportamiento elástico lineal, diafragma rígido por nivel mediante condensación de grados de libertad, empotramiento de la base, amortiguamiento del 5 % del crítico conforme a las NTC-Sismo (2020) para estructuras de concreto en régimen elástico, y excitación sísmica mediante el acelerograma SCT2 del sismo del 19 de septiembre del 2017, registrado por la Red Acelerográfica del CENAPRED. La comparación se distingue en tres niveles: análisis modal, modal espectral e historia de tiempo.

3.2.1. BASE RÍGIDA

La Tabla 3.1 presenta los períodos naturales de los tres primeros modos de vibrar para el modelo con la base rígida. El modo 1 corresponde a la traslación en dirección Y con masa participante del 80.52 % en la misma dirección, el modo 2 a traslación en dirección X con 80.53 % y el modo 3 a un modo predominantemente torsional con participación traslacional nula en ambas direcciones horizontales. La comparación se hace en los tres primeros modos de vibrar dado que el método de Superposición de Avilés López et al. modifica únicamente el período fundamental mediante la adición de la flexibilidad de la cimentación, siendo los modos superiores independientes de la interacción suelo-estructura en la formulación adoptada (NTC-Sismo 2020).

Cuadro 3.1: Análisis modal - Base rígida

Modo	T MATLAB (s)	T SAP2000 (s)	Error (%)
1	2.4254	2.4232	0.09
2	2.2950	2.3024	0.32
3	2.0382	2.1473	5.08

Las Figuras 3.6 y 3.7 muestran la forma modal del modo 1 obtenida en SAP2000, con traslación predominante en dirección Y .

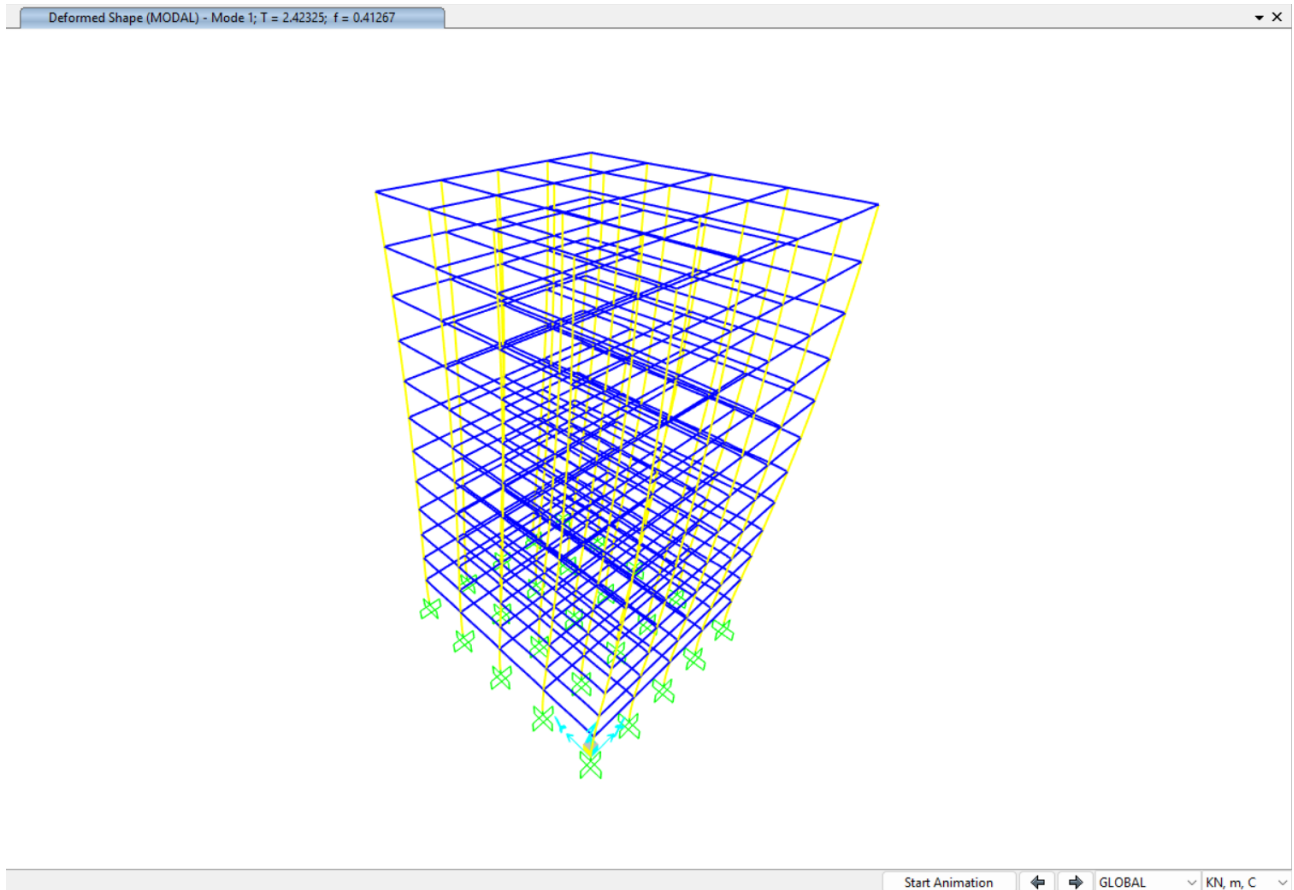


Figura 3.6: Forma modal del modo 1 en SAP2000, base rígida. Período $T_1 = 2.423$ s, frecuencia $f_1 = 0.413$ Hz. La deformada muestra traslación predominante en dirección Y con amplitud máxima en la azotea.

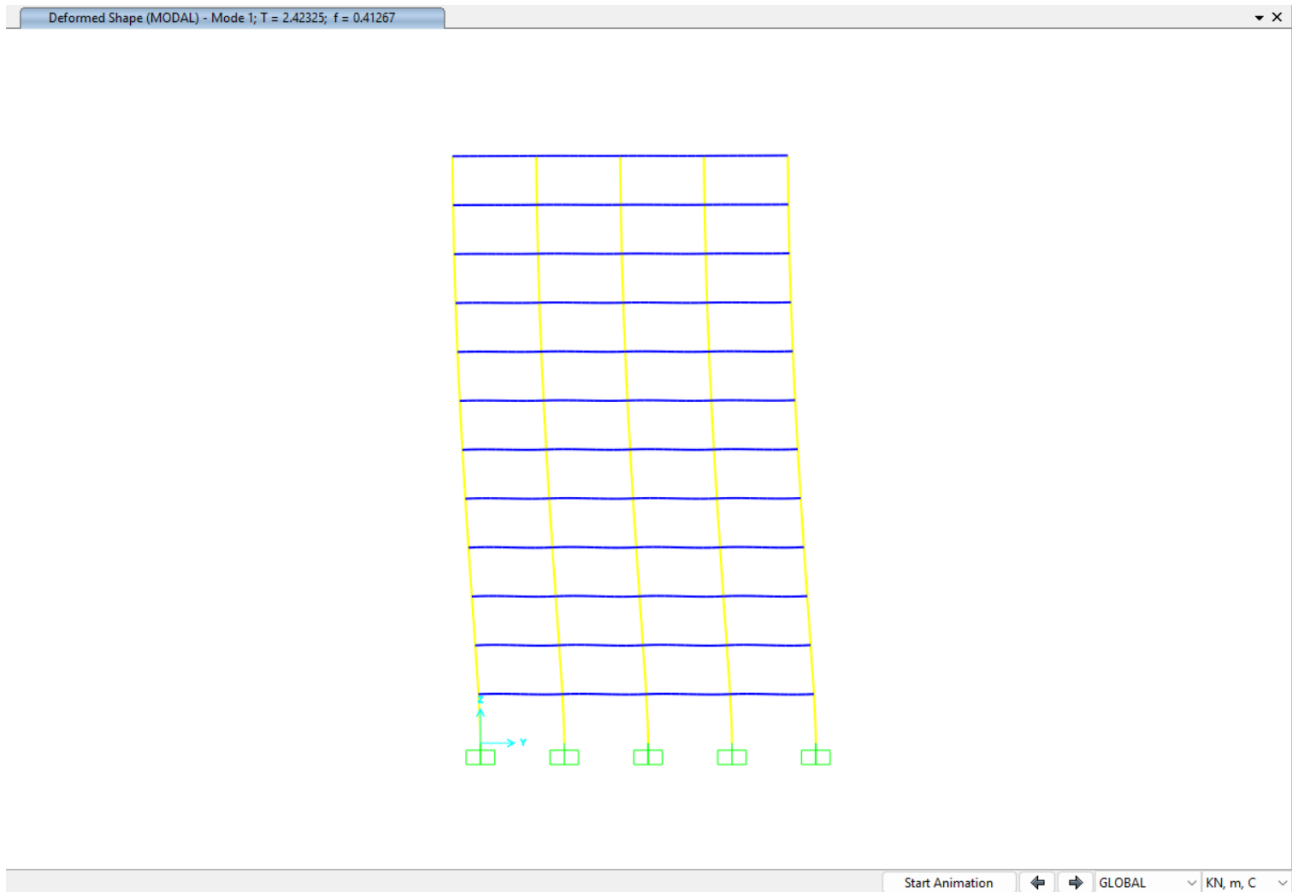


Figura 3.7: Forma modal del modo 1 en SAP2000, base rígida, vista en el plano $Y-Z$. Se aprecia el perfil de desplazamientos laterales tipo ménsula con curvatura creciente hacia los niveles superiores, característico del modo fundamental de traslación.

Los modos 1 y 2 presentan errores menores al 0.5 %, lo que indica que la distribución de masa y rigidez lateral del modelo generado por el script en MATLAB reproduce con precisión la del modelo de SAP2000. Con esta precisión es posible inferir que el ensamblaje de la matriz de rigidez global, mediante el método de la matriz de rigidez directa, la tributación nodal de masas de vigas, columnas y losas, así como la condensación del diafragma rígido producen un sistema dinámico equivalente al generado por SAP2000 con sus rutinas de ensamblaje y condensación.

El modo 3 presenta un error del 5.08 %. Este modo es predominantemente torsional, por lo que su período tiene mayor sensibilidad a la distribución de masa rotacional por nivel que los modos traslacionales. La discrepancia se atribuye a diferencias en el cómputo de la inercia de masa torsional del diafragma: en el modelo de MATLAB, la inercia rotacional de cada nodo se estima a partir de la sección transversal de los elementos convergentes, mientras que SAP2000 la calcula integrando la distribución espacial de masa en el plano del piso. Esta diferencia afecta con mayor proporción al modo torsional que a los traslacionales, generando un período del modo 3 ligeramente mayor en MATLAB que en SAP2000. La discrepancia no afecta el análisis de Interacción Suelo-Estructura, que opera de manera exclusiva sobre el modo traslacional dominante (AvilesLopez2016; Cook et al., 1989).

(Avilés López et al., 2016; Cook et al., 1989).

3.2.2. PERÍODOS NATURALES - BASE FLEXIBLE (INTERACCIÓN SUELO-ESTRUCTURA)

La Tabla 3.2 presenta la comparación de períodos obtenidos tomando en cuenta la Interacción Suelo-Estructura (ISE). El período efectivo \tilde{T} se obtiene de la suma cuadrática de los períodos asociados a la estructura con base rígida (T_e), a la flexibilidad traslacional de la cimentación (T_b) y al cabeceo (T_r), conforme a las ecuaciones de las NTC-Sismo (2020), como se muestra en la siguiente expresión.

$$\tilde{T}^2 = T_e^2 + T_b^2 + T_r^2 \quad (3.2)$$

Las impedancias de la cimentación fueron calculadas con las formulaciones de Gazetas para disco circular rígido sobre un semiespacio elástico homogéneo con los parámetros del suelo mencionados en la Tabla 2.1

y conforme a las NTC-Cimentaciones (2017). Las rigideces estáticas resultantes fueron $K_b = 5.27 \times 10^8$ N/m para la traslación horizontal y $K_r = 8.35 \times 10^{10}$ N·m/rad para el cabeceo.

Cuadro 3.2: Comparación de períodos - Modelo con ISE

Modo	T MATLAB base rígida (s)	\tilde{T} MATLAB ISE (s)	T SAP2000 ISE (s)	Error (%)
1	2.4254	2.7804	2.7858	0.19
2	2.2950	2.2950	2.3024	0.32
3	2.0382	2.0382	2.1473	5.08

El porcentaje de error del 0.19 % en el período ISE fundamental constituye el resultado más importante de este trabajo. La magnitud del error es incluso menor que el error del período con base rígida entre MATLAB y SAP2000 (0.09 %), lo que indica que el proceso iterativo de cálculo de \tilde{T} no acumula un error numérico significativo. La convergencia del algoritmo iterativo se alcanzó en dos iteraciones con una tolerancia de 10^{-6} , confirmando la estabilidad numérica del procedimiento (NTC-Sismo 2020). Los modos 2 y 3 no se modifican por el análisis ISE en la formulación de Avilés López et al., 2016, de manera que sus errores respecto a SAP2000 son idénticos a los de la Tabla 3.1.

3.2.3. DESPLAZAMIENTOS EN AZOTEA

La Tabla 3.3 presenta los desplazamientos máximos en el último nodo de la azotea para el análisis espectral e historia de tiempo en dirección Y, que es la dirección de excitación del acelerograma SCT2-2017 con componente Norte-Sur. En el análisis espectral se aplicó la combinación cuadrática SRSS sobre los primeros 12 modos calculados, con las aceleraciones espectrales extraídas del espectro de diseño de SASID para Zona III - Lago con factor de comportamiento sísmico de $Q = 2$. Para el análisis de historia de tiempo la integración se realizó mediante el método de Newmark- β con parámetros $\gamma = 0.5$ y $\beta = 0.25$, correspondientes al esquema de aceleración promedio constante, incondicionalmente estable y sin amortiguamiento numérico artificial (Hori, 2011).

Cuadro 3.3: Desplazamiento máximo en azotea - Dirección Y

Caso	MATLAB (cm)	SAP2000 (cm)	Error (%)
Espectral (SRSS)	29.61	29.29	1.08
Historia de tiempo	15.48	15.24	1.55

La Figura 3.8 muestra la deformada del modelo bajo la carga estática de rigidez lateral en dirección Y, utilizada para verificar la consistencia del modelo antes del análisis dinámico.

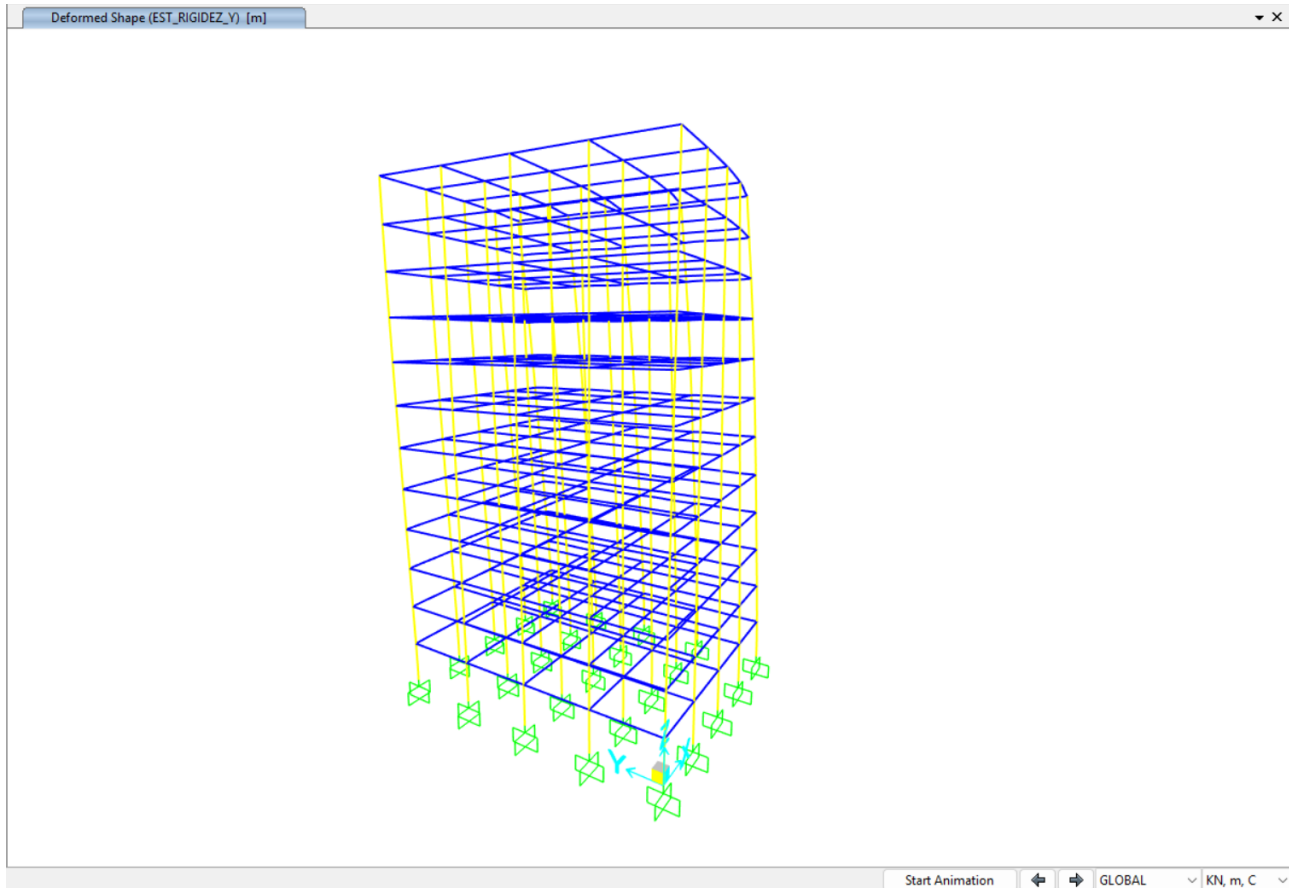


Figura 3.8: Deformada del modelo bajo carga estática lateral en dirección Y (caso EST_RIGIDEZ_Y) en SAP2000. La forma de la deformada es consistente con la distribución de rigidez lateral del sistema de marcos en esa dirección.

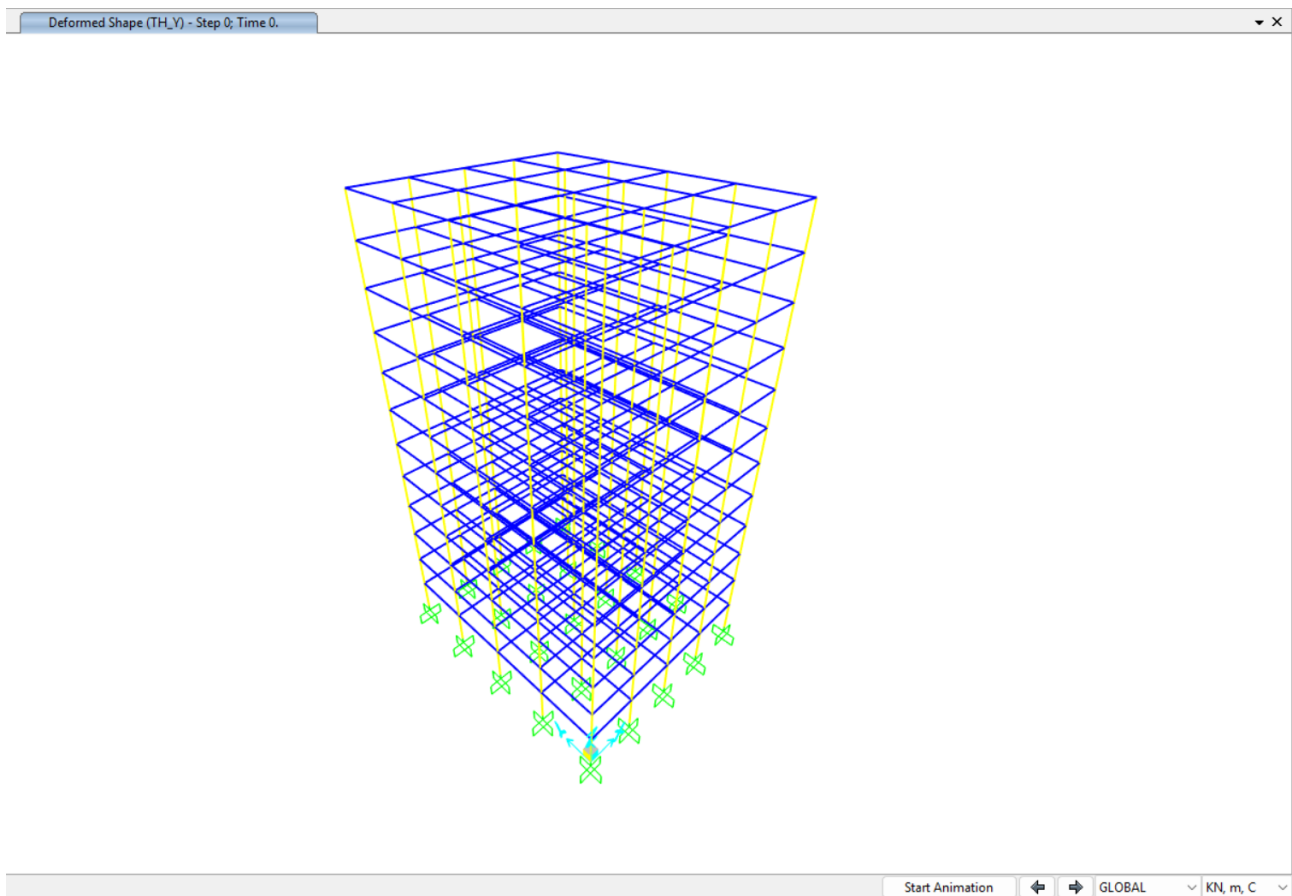


Figura 3.9: Estado inicial de la estructura en el análisis tiempo-historia en dirección Y (caso TH_Y, $t = 0$ s) en SAP2000. La estructura se encuentra en reposo antes de la aplicación del acelerograma SCT2-2017.

El porcentaje de error es inferior al 2 % en ambos casos. Esta concordancia confirma que el modelo reproduce con precisión las propiedades dinámicas y la respuesta ante excitación sísmica real, lo que involucra la superposición de todos los modos activos, la integración temporal paso a paso y la reconstrucción de desplazamientos físicos a partir de coordenadas modales. Los errores son del mismo orden en ambos análisis, lo que permite descartar la presencia de errores sistemáticos en la implementación de la transformación modal o en la reconstrucción de desplazamientos a partir de los grados de libertad condensados (Chopra, 2007).

El desplazamiento espectral (29.61 cm) es aproximadamente el doble del desplazamiento máximo de historia de tiempo (15.48 cm). La diferencia no representa una inconsistencia: el análisis espectral utiliza la envolvente máxima del espectro de diseño proporcionado por SASID, que incorpora un factor de reducción por comportamiento sísmico de $Q = 2$ aplicado sobre la demanda elástica (NTC-Sismo, 2020). El análisis de

historia de tiempo, en cambio, utiliza un acelerograma real cuya intensidad en el rango de períodos del edificio ($T \approx 2.4$ s) es inferior a la demanda espectral de diseño, como se verifica al comparar la aceleración máxima del registro SCT2-2017 ($a_{max} = 0.09g$) con la ordenada espectral de diseño en ese período ($Sa = 0.157g$ para $Q = 2$). La relación entre estos dos análisis es esperada y no requiere corrección adicional (Chopra, 2007; Gobierno de la Ciudad de México, 2017).

3.3. ANÁLISIS DE DIFERENCIAS EN COMPORTAMIENTO ESTRUCTURAL

3.3.1. EFECTO DE LA INTERACCIÓN SUELO-ESTRUCTURA SOBRE EL PERÍODO FUNDAMENTAL

Las Figuras 3.10, 3.11 y 3.12 muestran la forma modal del modo 1 en el modelo ISE de SAP2000 ($\tilde{T} = 2.786$ s). En comparación con la forma modal de base rígida (Figuras 3.6 y 3.7), se aprecia que la deformada con base flexible incorpora una rotación en la base debida a la flexibilidad rotacional del suelo, lo cual es físicamente consistente con la contribución dominante del cabeceo (83.6 %) al incremento total del período.

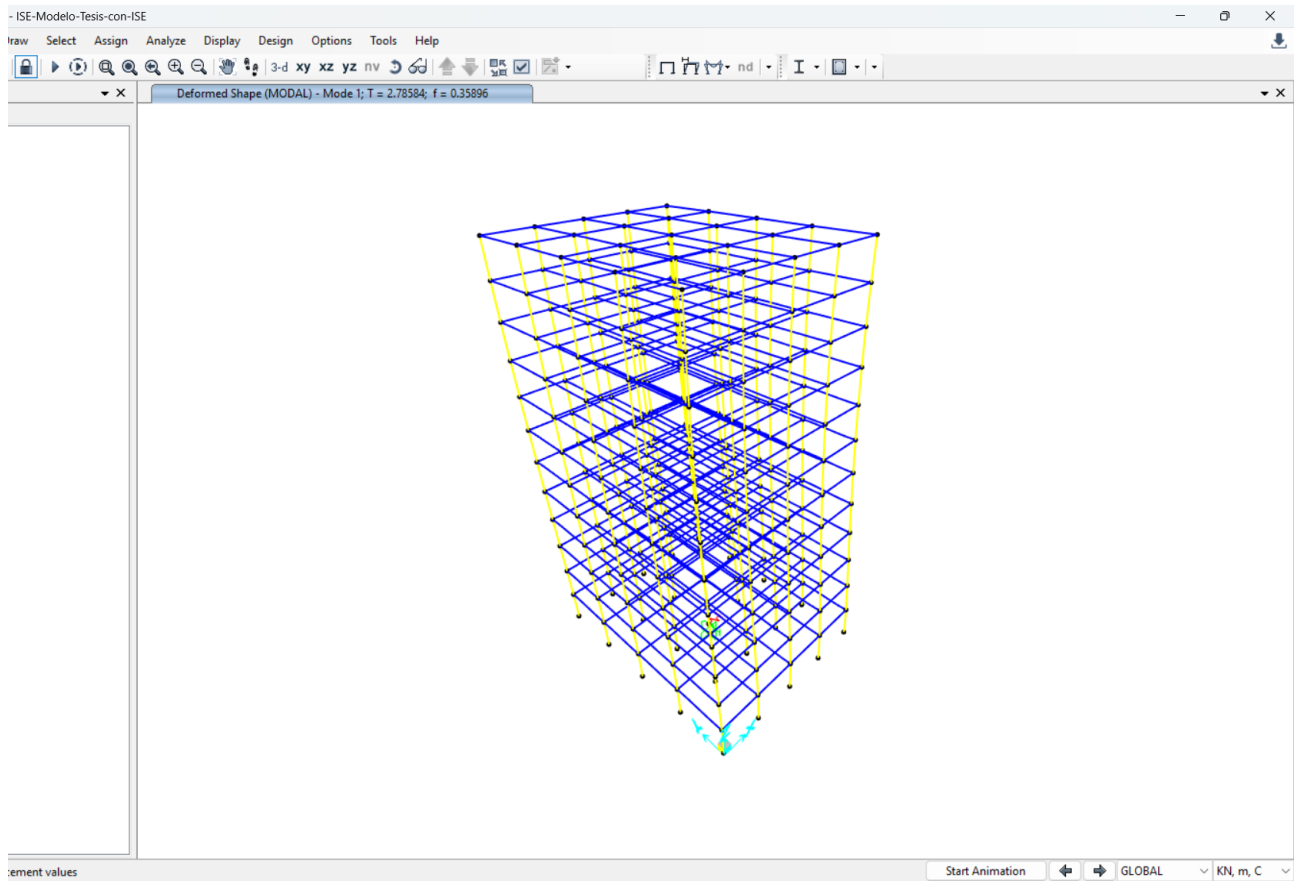


Figura 3.10: Forma modal del modo 1 en el modelo ISE de SAP2000, vista tridimensional frontal. Período $\tilde{T}_1 = 2.786$ s, frecuencia $\tilde{f}_1 = 0.359$ Hz. La deformada incorpora traslación de la base y rotación por cabeceo, que no están presentes en el modelo de base rígida.

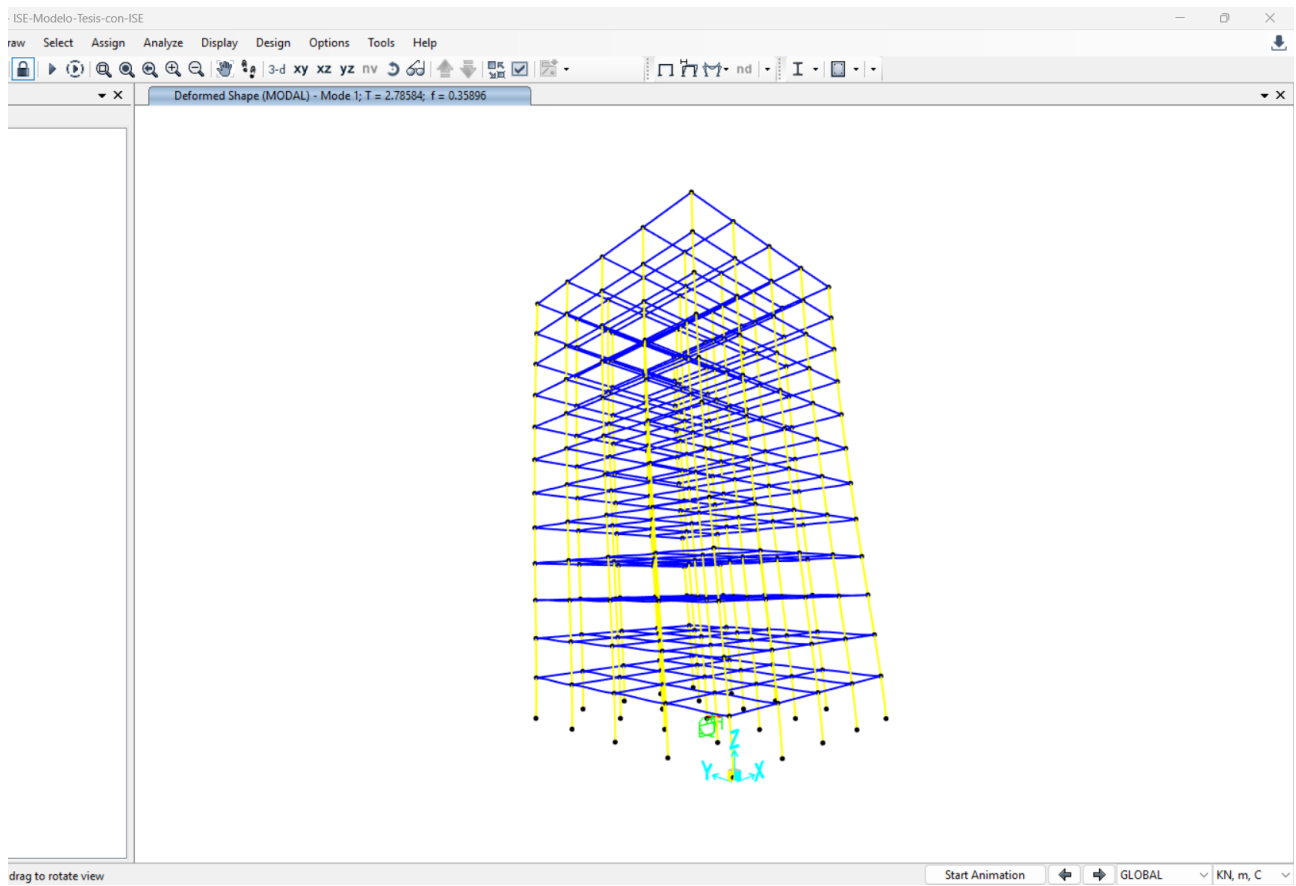


Figura 3.11: Forma modal del modo 1 en el modelo ISE de SAP2000, vista tridimensional lateral. Se observa la inclinación de la base respecto a la vertical, que refleja la deformación rotacional del resorte de cabeceo K_r .

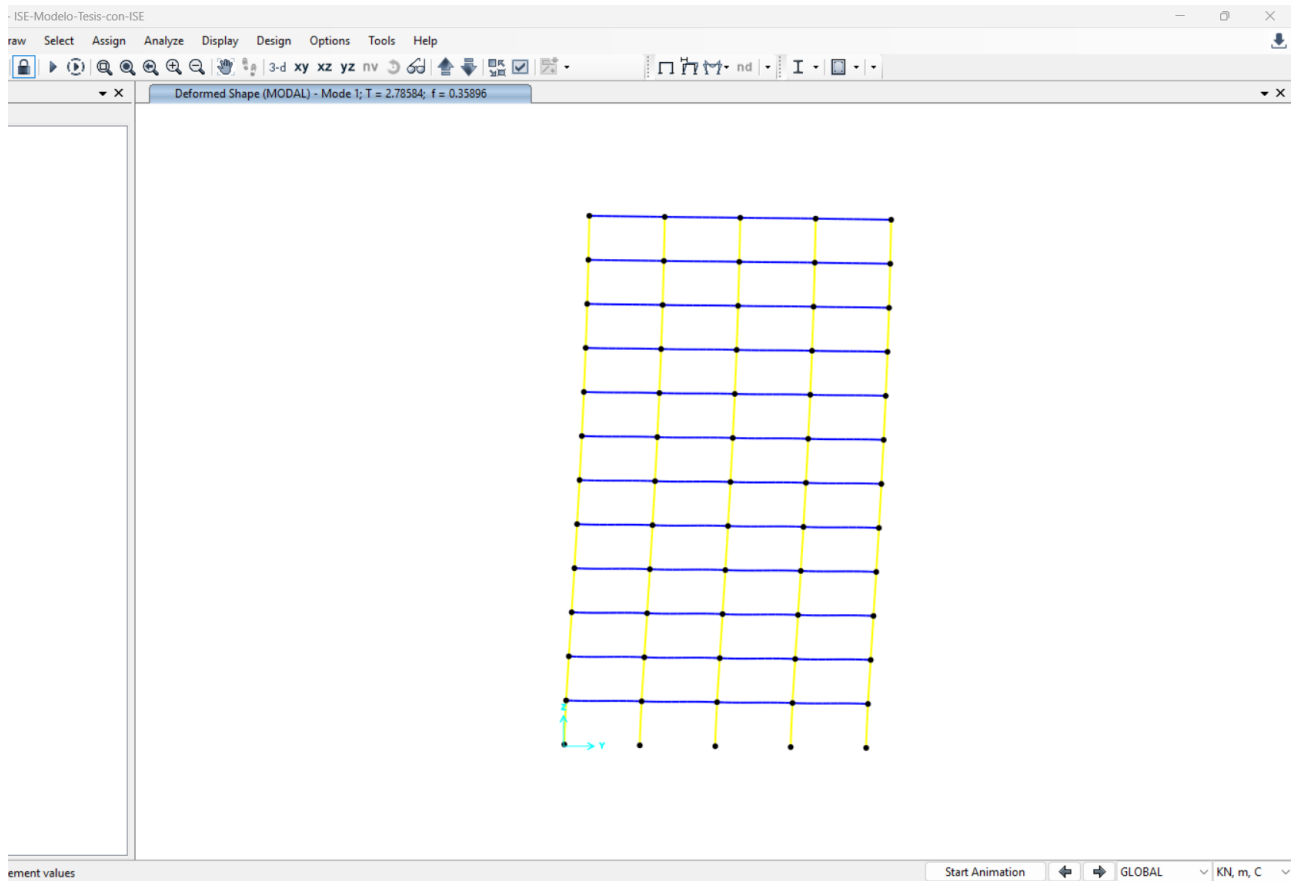


Figura 3.12: Forma modal del modo 1 en el modelo ISE de SAP2000, vista en el plano $Y-Z$. La rotación en la base es visible en la inclinación de las columnas del primer nivel; el perfil de desplazamientos resultante combina la deformación de la superestructura con la traslación y el cabeceo de la cimentación.

La incorporación de flexibilidad a la cimentación produjo un incremento del período fundamental de 14.6 %, al pasar de $T_1 = 2.425$ s con base rígida a $\tilde{T} = 2.780$ s con base flexible. Para interpretar adecuadamente este resultado es necesario examinar la descomposición del incremento en sus componentes traslacional y rotacional. Los períodos parciales calculados en MATLAB fueron $T_b = 0.550$ s para la traslación horizontal de la base y $T_r = 1.243$ s para el cabeceo.

Las contribuciones al incremento total $\tilde{T}^2 - T_e^2$ se calculan al sustituir en la Ec. 3.2:

$$\tilde{T}^2 - T_e^2 = (2.780)^2 - (2.425)^2 \quad (3.3)$$

$$\tilde{T}^2 - T_e^2 = 7.7284 - 5.8806 \quad (3.4)$$

$$\tilde{T}^2 - T_e^2 = 1.8478 \text{ s}^2 \quad (3.5)$$

Este incremento total se descompone en las contribuciones de la traslación horizontal y del cabeceo de la cimentación.

- Contribución por traslación horizontal

$$T_b^2 = (0.550)^2 \quad (3.6)$$

$$T_b^2 = 0.3025 \text{ s}^2 \quad (3.7)$$

$$\frac{T_b^2}{\tilde{T}^2 - T_e^2} \times 100 \quad (3.8)$$

$$\frac{0.3025}{1.8478} \times 100 = 16.4 \% \quad (3.9)$$

- Contribución por cabeceo

$$T_r^2 = (1.243)^2 \quad (3.10)$$

$$T_r^2 = 1.5453 \text{ s}^2 \quad (3.11)$$

$$\frac{T_r^2}{\tilde{T}^2 - T_e^2} \times 100 \quad (3.12)$$

$$\frac{1.5453}{1.8478} \times 100 = 83.6 \% \quad (3.13)$$

De esta manera se obtienen las contribuciones por traslación horizontal de 16.4 % y por cabeceo de 83.6 %.

Con estos resultados se muestra que la losa de cimentación (24 m × 20 m) se relaciona con el período T_b a través de la expresión correspondiente al oscilador traslacional equivalente de la base (Avilés López et al.,

2016):

$$T_b = 2\pi \sqrt{\frac{M^*}{K_b}} \quad (3.14)$$

$$T_b = 2\pi \sqrt{\frac{4,033,900}{5.27 \times 10^8}} \quad (3.15)$$

$$T_b = 0.550 \text{ s} \quad (3.16)$$

Por otro lado, el período asociado al cabeceo de la cimentación se obtiene de la siguiente manera:

$$T_r = 2\pi \sqrt{\frac{M^* H_c^2}{K_r}} \quad (3.17)$$

$$T_r = 2\pi \sqrt{\frac{4,033,900 \times (28.47)^2}{8.35 \times 10^{10}}} \quad (3.18)$$

$$T_r = 1.243 \text{ s} \quad (3.19)$$

Dado que $T_b = 0.550 \text{ s} \ll T_r = 1.243 \text{ s}$, el cabeceo domina el incremento de período por ISE. Esto está relacionado a la geometría de la cimentación: una losa de $24 \text{ m} \times 20 \text{ m}$ produce una rigidez traslacional K_b elevada que restringe el movimiento horizontal de la base, mientras que la rigidez rotacional K_r , aunque mayor en valor absoluto, actúa sobre un brazo de palanca igual a $H_c = 28.47 \text{ m}$, lo que amplifica el efecto de la flexibilidad rotacional sobre la respuesta del sistema (Avilés López et al., 2016; Gazetas, 1991).

En el espectro de diseño SASID para la Zona III - Lago, el período $\tilde{T} = 2.780 \text{ s}$ se ubica en la rama descendente del espectro, donde la aceleración espectral disminuye al aumentar el período. Por tanto, el alargamiento de período producido por la ISE reduce la aceleración espectral de diseño respecto a la que correspondería con base rígida, lo que es consistente con la reducción del cortante basal permitida al incorporar efectos de ISE en las NTC-Sismo (2020) (Avilés López et al., 2016; Pinto-Vega y Ledezma, 2019).

Cabe señalar que en estructuras con períodos cortos, ubicados en la rama ascendente del espectro, el alargamiento de período producido por la ISE incrementa la demanda sísmica en lugar de reducirla. Este efecto se ha documentado en edificios de baja altura en la Zona III (Avilés López et al., 2016; J. Mayoral et al., 2019).

Sin embargo, el incremento de 14.6% es inferior al rango reportado en la literatura para edificios de mediana altura en Zona III - Lago, que es $+20 \%$ a $+55 \%$ (Avilés López et al., 2016; J. Mayoral et al., 2019; Quinde y Reinoso, 2020). La diferencia se atribuye a la rigidez lateral elevada que ofrece una cimentación de amplia superficie.

El método supone suelo homogéneo y cimentación rígida, condiciones que son aproximaciones al compor-

tamiento real de los depósitos lacustres de la Zona III. La heterogeneidad estratigráfica real y la posible no linealidad del suelo bajo excitación intensa podrían modificar las impedancias de manera significativa respecto a los valores calculados con las formulaciones de Gazetas, lo que introduce una incertidumbre sobre el valor real del incremento de período que el modelo no puede capturar (NTC-Cimentaciones, 2017).

3.4. LIMITACIONES DEL CÓDIGO IMPLEMENTADO

El modelo implementado en MATLAB opera bajo supuestos cuyas implicaciones deben considerarse al interpretar los resultados. El análisis se restringe al régimen elástico lineal. En consecuencia, no se captura la degradación de rigidez, el agrietamiento del concreto ni la disipación histerética de energía bajo excitaciones de gran intensidad (Chopra, 2007). El diafragma rígido implementado mediante condensación *master-slave* impone un desplazamiento lateral uniforme en cada nivel, lo que impide capturar deformaciones en planta y efectos de torsión irregular (Cook et al., 1989). La Interacción Suelo-Estructura se modela con el método de superposición cuadrática de Avilés López et al., que asume suelo homogéneo, cimentación perfectamente rígida y respuesta lineal del depósito. Estas condiciones son aproximaciones respecto al comportamiento real de los depósitos lacustres de la Zona III - Lago, cuya estratigrafía es heterogénea y cuyo comportamiento dinámico es marcadamente no lineal bajo excitaciones intensas (NTC-Cimentaciones, 2017, J. Mayoral et al., 2019; J. M. Mayoral y Pérez, 2024).

El amortiguamiento se modela mediante coeficientes de Rayleigh calibrados para 5 % de amortiguamiento crítico en los modos 1 y 3, lo que produce valores de amortiguamiento efectivo variables en los modos intermedios y superiores (Hori, 2011). La excitación sísmica se aplica en una sola dirección horizontal (Y), sin considerar la componente vertical ni los efectos de incidencia oblicua del movimiento sísmico.

3.4.1. DIFICULTADES EN LA METODOLOGÍA OBSERVADA DEL MODELO ENTRE MATLAB Y SAP2000

Durante el proceso de validación se identificaron diferencias en la forma en que cada plataforma maneja la masa de losas y la restricción del diafragma rígido, cuya resolución fue condición necesaria para que la comparación de períodos tuviera sustento físico.

En MATLAB la masa de las losas se calcula explícitamente mediante tributación de área por densidad superficial (600 kg/m^2) y se distribuye directamente a los nodos de cada nivel a través de la matriz de masa ensamblada, resultando en una masa total de 5,036 ton que incluye la contribución de columnas, vigas y losas (Cook et al., 1989; Hori, 2011). Por otro lado, SAP2000 calcula automáticamente la masa de los elementos `frame` a partir de su peso propio mediante la opción *Element Self Mass*. Sin embargo, al no existir losas modeladas como elementos físicos, su masa no se generaba de manera automática. Para reproducir la distribución de masa de MATLAB fue necesario convertir la masa de losas a una carga distribuida equivalente de 12.84 kN/m sobre las vigas de cada nivel y activarla como masa mediante *Specified Load Pattern* $\times 1.0$ en la fuente de masa del modelo de SAP2000.

El modelo ISE en SAP2000 se construyó de manera independiente al modelo de base rígida. En lugar de empotramientos en la base, se asignaron resortes lineales equivalentes en el nodo central de la planta de cimentación, con rigideces K_b y K_r , calculadas conforme a las expresiones de Gazetas. Las Figuras 3.13, 3.14 y 3.15 muestran la configuración del modelo ISE en SAP2000.

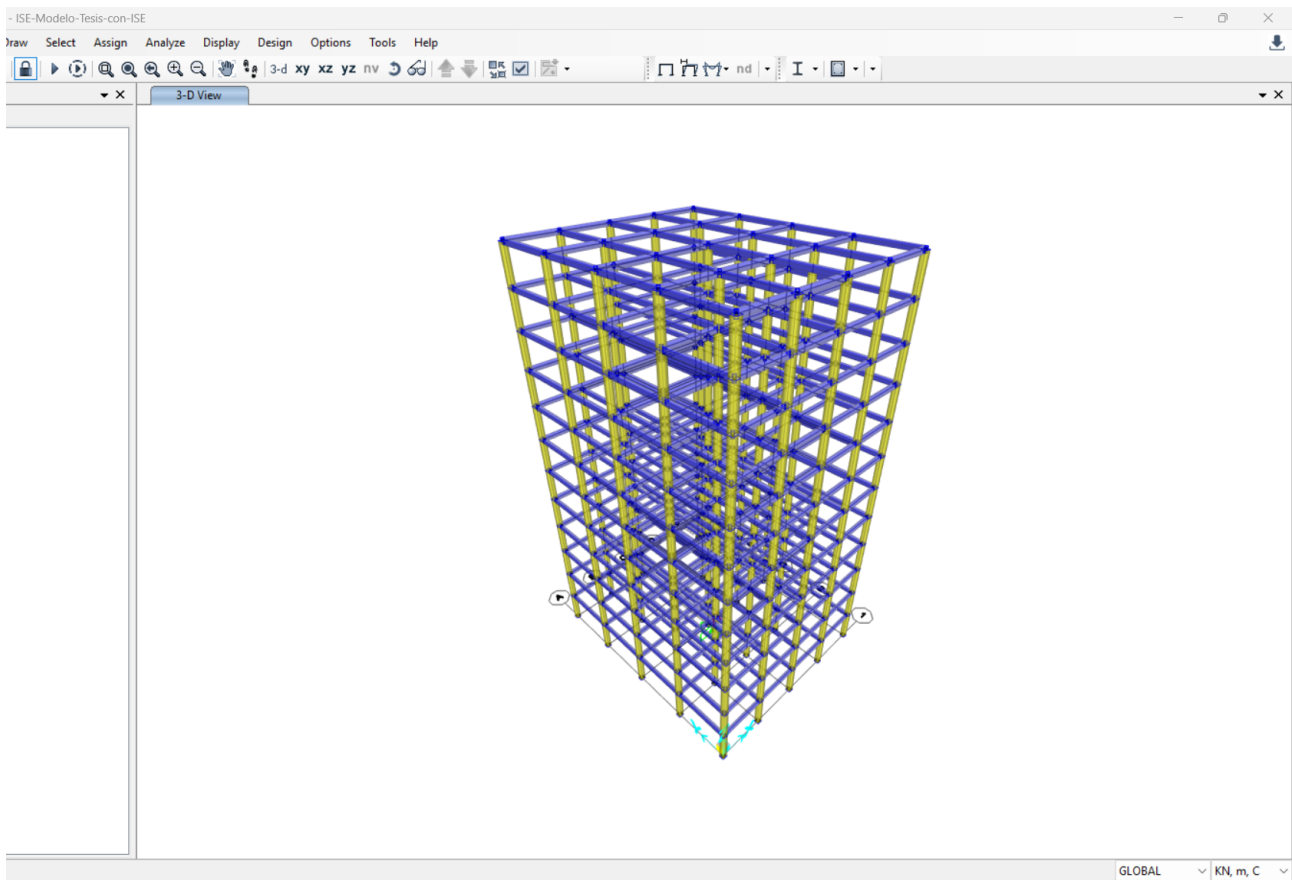


Figura 3.13: Vista tridimensional del modelo ISE en SAP2000. La ausencia de símbolos de empotramiento en la base distingue este modelo del de base rígida; en su lugar se asignan resortes lineales equivalentes en el nodo central de la cimentación.

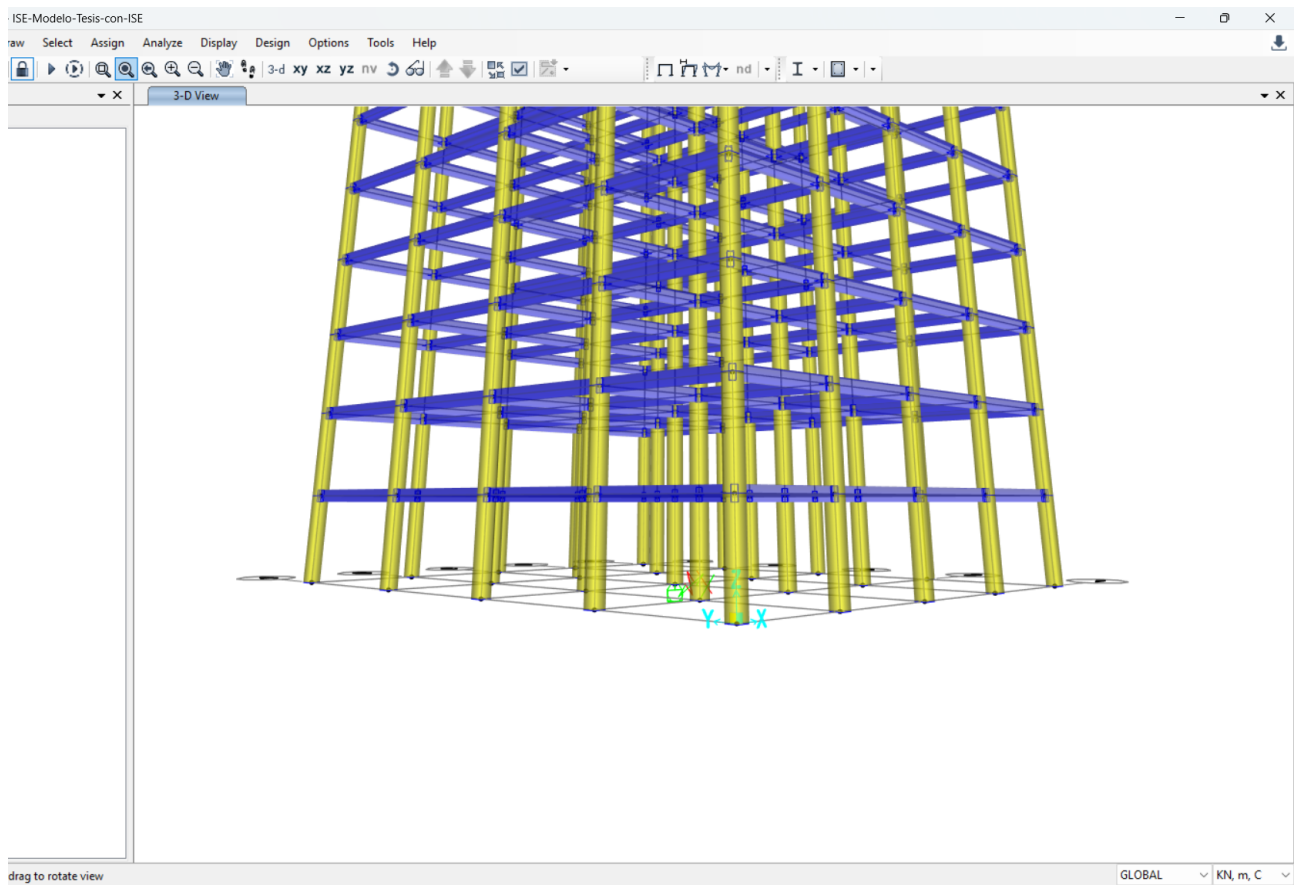


Figura 3.14: Detalle de la zona de cimentación del modelo ISE en SAP2000. Las columnas descansan sobre nodos libres en $z = 0$, conectados al nodo central mediante vínculos rígidos que concentran la flexibilidad del suelo en un único punto de aplicación de los resortes equivalentes.

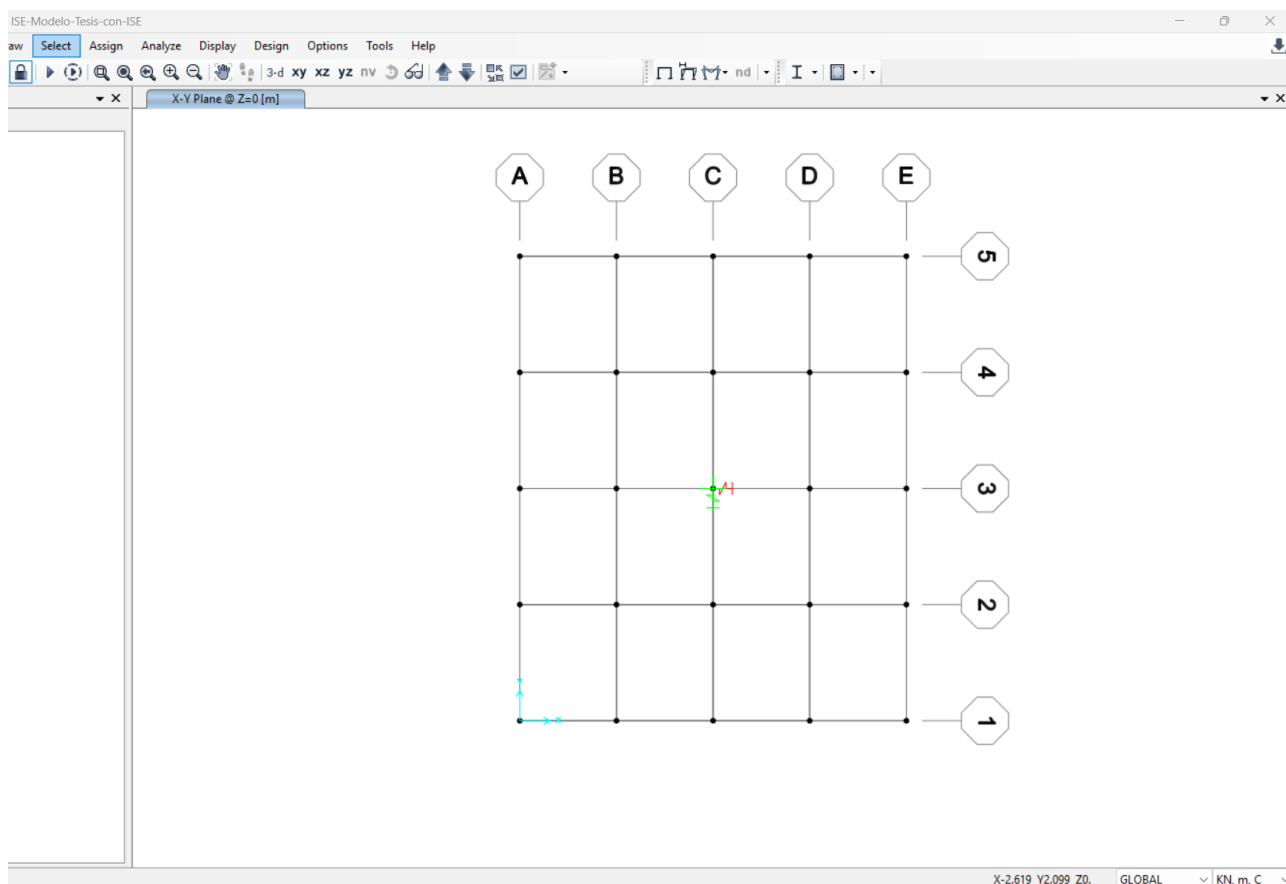


Figura 3.15: Planta del modelo ISE en SAP2000 a nivel $z = 0$. El nodo central (marcado en verde) concentra las rigideces equivalentes del suelo K_b y K_r ; los nodos periferales de la base están vinculados a él mediante restricciones rígidas que reproducen el comportamiento de la losa de cimentación como cuerpo rígido.

En una versión inicial del modelo de SAP2000 se intentó modelar las losas como elementos `shell` de membrana para que el programa generara su masa automáticamente. Sin embargo, los elementos `shell` de membrana son físicamente incompatibles con la restricción de diafragma rígido: la condición de diafragma elimina los desplazamientos relativos en el plano de cada nivel, que son justamente los grados de libertad que activan la rigidez de membrana del elemento. Al anularse estos desplazamientos relativos, los `shells` no contribuían a la rigidez del modelo y producían períodos fundamentales incorrectos (Cook et al., 1989). Esta incompatibilidad entre elementos `shell` de membrana y diafragma rígido está documentada en la teoría de elementos finitos: la rigidez de membrana requiere deformación relativa en el plano para generar fuerzas internas, condición que el diafragma rígido elimina por definición (Cook et al., 1989).

Esta incompatibilidad tiene una consecuencia directa sobre la verificación normativa que proponen las NTC-Sismo (2020), que establecen como criterio simplificado que la Interacción Suelo-Estructura es despreciable

cuando la rigidez lateral del suelo es al menos 2.5 veces mayor que la rigidez lateral de la estructura, verificación que depende del período fundamental del modelo. Si el período fundamental está mal calculado por una masa o rigidez incorrecta, el cociente de rigideces resulta erróneo y la verificación del umbral normativo pierde validez. La corrección del modelo, que consistió en eliminar los `shells` e igualar la distribución de masa entre MATLAB y SAP2000 mediante carga distribuida equivalente, fue condición necesaria para que tanto la comparación de períodos como la verificación normativa tuvieran sustento físico.

3.5. INTERPRETACIÓN DE LOS RESULTADOS Y FIABILIDAD DEL MODELO

Los errores obtenidos en la comparación entre MATLAB y SAP2000 son inferiores al 5.1 % en períodos modales y al 2 % en desplazamientos máximos para todos los casos analizados. El error en el período ISE fundamental, parámetro central del método de Avilés López et al., es de 0.19 %. Todos estos valores se encuentran dentro de los márgenes de aceptación reportados en la literatura para validaciones de modelos numéricos de marcos tridimensionales de concreto reforzado (Chopra, 2007; Cook et al., 1989).

Los resultados son válidos dentro del alcance declarado en la sección 0.7: análisis sísmico elástico lineal de un edificio de concreto reforzado de 12 niveles en Zona III del Valle de México, con interacción suelo-estructura modelada mediante el método de superposición cuadrática conforme a las NTC-Sismo (2020). Fuera de estos supuestos, los resultados no deben extrapolarse sin revisar las limitaciones descritas en la sección anterior.

4

Conclusiones

Este trabajo implementó en MATLAB el método de superposición de Avilés López et al. para un edificio de concreto reforzado de 12 niveles en Zona III del Valle de México, incorporando las impedancias de Gazetas como rigideces equivalentes en los nodos de la base. La comparación sistemática contra SAP2000 v26.3.0 arrojó un error de 0.19 % en el período ISE fundamental, que es el parámetro central del método. El algoritmo iterativo alcanzó convergencia en dos iteraciones con tolerancia 10^{-6} , resultado consistente con los criterios de estabilidad numérica establecidos en las NTC-Sismo (2020).

El período fundamental pasó de $T_1 = 2.425$ s con base rígida a $\tilde{T} = 2.780$ s con base flexible, un incremento de 14.6 %. Descompuesto en sus componentes el cabeceo explica el 83.6 % del incremento total $\tilde{T}^2 - T_c^2$ y la traslación horizontal el 16.4 % restante. La dominancia del cabeceo se explica por el cociente $T_r^2 / (T_b^2 + T_r^2)$:

la losa de $24 \text{ m} \times 20 \text{ m}$ genera una rigidez traslacional $K_b = 5.27 \times 10^8 \text{ N/m}$ que mantiene $T_b = 0.550 \text{ s}$ relativamente pequeño, mientras que la rigidez rotacional $K_r = 8.35 \times 10^{10} \text{ N}\cdot\text{m/rad}$ actúa sobre una altura efectiva $H_e = 28.47 \text{ m}$ que amplifica la contribución al período de cabeceo $T_r = 1.243 \text{ s}$.

En cuanto a los períodos modales de base rígida, los errores respecto a SAP2000 fueron 0.09% en el modo 1 y 0.32% en el modo 2. El modo 3 presentó 5.08% , valor atribuible a diferencias en el cómputo de la inercia de masa torsional entre ambas plataformas. Dado que el método de superposición empleado en el análisis ISE considera únicamente el modo fundamental, el error del modo 3 no incide en los resultados de desplazamiento.

Para los desplazamientos máximos en azotea, los errores fueron 1.08% en el análisis espectral y 1.55% en el análisis de historia de tiempo. Ambos valores se encuentran dentro del rango de $\pm 5 \%$ reportado como aceptable en validaciones de marcos tridimensionales de concreto reforzado (Chopra, 2007; Cook et al., 1989).

El incremento de 14.6% es menor que el rango de $+20 \%$ a $+55 \%$ documentado para edificios de mediana altura en Zona III (Avilés López et al., 2016; J. Mayoral et al., 2019; Quinde y Reinoso, 2020), diferencia que se explica por la alta rigidez que ofrece una cimentación de gran superficie en planta. Los supuestos de suelo homogéneo y cimentación rígida del método de Gazetas introducen incertidumbre sobre el valor real del incremento, dado que los depósitos lacustres de la Zona III presentan estratigrafía heterogénea y comportamiento no lineal bajo excitaciones intensas (NTC-Cimentaciones, 2017).

Durante la validación se identificó una incompatibilidad física entre los elementos *shell* de membrana y la restricción de diafragma rígido en SAP2000: al eliminarse los desplazamientos relativos en el plano por el diafragma, los *shells* no activaban su rigidez de membrana y producían períodos incorrectos. En el modelo de MATLAB el diafragma rígido está implementado mediante condensación cinemática master-slave, lo que produce igualmente deformación relativa nula entre los nodos de cada nivel, anulando la contribución efectiva de la membrana a la rigidez del sistema. La corrección en SAP2000 consistió en representar la masa de losas como carga distribuida equivalente de 12.84 kN/m sobre las vigas, activada como fuente de masa, lo que igualó la distribución de masa entre los resultados de SAP2000 y MATLAB e hizo válida la comparación

de períodos (Cook et al., 1989).

Bibliografía

- Ángel, M., García Illescas, M., Baruo, D., Sánchez Aldama, A., Pérez, D., Vila, D., & Ramírez-Guzmán, L. (2024). Efectos de interacción suelo–estructura estimados con métodos paramétricos. *XXIV Congreso Nacional de Ingeniería Sísmica: Hacia la Resiliencia Sísmica de México*.
- Avilés, J., & Pérez-Rocha, L. E. (1998). Site effects and soil-structure interaction in the valley of Mexico. *Soil Dynamics and Earthquake Engineering*, 17(1), 29-39. [https://doi.org/https://doi.org/10.1016/S0267-7261\(97\)00027-4](https://doi.org/https://doi.org/10.1016/S0267-7261(97)00027-4)
- Avilés, J., & Pérez-Rocha, L. E. (2010). Regional subsidence of Mexico City and its effects on seismic response. *Soil Dynamics and Earthquake Engineering*, 30(10), 981-989. <https://doi.org/https://doi.org/10.1016/j.soildyn.2010.04.009>
- Avilés López, J., Demeneghi Colina, A., López Rincón, G., Pérez Rocha, L. E., Sánchez Sesma, J. F., Suárez López, M. M., & Trigos Suárez, J. L. (2016). *Interacción Suelo-Estructura, Estática y Dinámica*. Sociedad Mexicana de Ingeniería Geotécnica.
- Ayse Hortacsu, M. A. E., Ayhan Irfanoglu, & Aydın, M. F. (2023). Advance Reconnaissance Team Observations of Earthquake Impacts in Turkey. *Earthquake Engineering Research Institute (EERI)*. <https://eeri.org/about-eeri/news/15649-lfe-advance-reconnaissance-team-observations-of-earthquake-impacts-in-turkey>
- Barnaure, M., & Manoli, D. (2019). Unfavourable seismic behaviour of reinforced concrete structures due to soil structure interaction. *IOP Conference Series: Earth and Environmental Science*, 362(1), 012119. <https://doi.org/10.1088/1755-1315/362/1/012119>
- Celebi, M., Prince, J., Dietel, C., Onate, M., & Chavez, G. (1987). The Culprit in Mexico City—Amplification of Motions. *Earthquake Spectra*, 3(2), 315-328. <https://doi.org/10.1193/1.1585431>
- Celigüeta-Lizarza, J. T. (2011). Método de los Elementos Finitos para Análisis Estructural. *Universidad de Navarra*. <https://hdl.handle.net/10171/19069>

- Chávez-García, G. A., Francisco J y Montalva. (2014). Site effects for Geotechnical Engineers, case study at Parkway valley. *Obras y proyectos*, 6-30.
- Chopra, A. (2007). *Dynamics of Structures*. Pearson Education. <https://books.google.com.mx/books?id=0dU1bDaRyP4C>
- Cook, R. D., Malkus, D. S., & Plesha, M. E. (1989). *Concepts and Applications of Finite Element Analysis* (3.^a ed.). John Wiley & Sons.
- Cruz-Atienza, V. M., Tago, J., Sanabria-Gómez, J. D., Chaljub, E., Etienne, V., Virieux, J., & Quintanar, L. (2016). Long Duration of Ground Motion in the Paradigmatic Valley of Mexico. *Scientific Reports*, 6(1), 38807. <https://doi.org/10.1038/srep38807>
- Dao, T. A., Mattsson, K., & Nazarov, M. (2022). Energy stable and accurate coupling of finite element methods and finite difference methods. *Journal of Computational Physics*, 449, 110791. <https://doi.org/https://doi.org/10.1016/j.jcp.2021.110791>
- Diaz-Fanas, G., Garini, E., Ktenidou, O.-J., Gazetas, G., Vaxevanis, T., Lan, Y. J., Heintz, J., Ma, X., Korre, E., Valles-Mattox, R., Stavridis, A., Kim, I., Hernandez-Bassal, L., Anzola, E., Berkowitz, R., Hussain, S., Jalalian, A., Carrion, H., Maldonado, V. D., ... Nikolaou, S. (2020). ATC Mw7.1 Puebla–Morelos earthquake reconnaissance observations: Seismological, geotechnical, ground motions, site effects, and GIS mapping. *Earthquake Spectra*, 36(2_suppl), 5-30. <https://doi.org/10.1177/8755293020964828>
- Diego Melgar, L. Q., V. J. Sahakian, & Çelebi, M. (2018). The 19 September 2017 M 7.1 Puebla-Morelos Earthquake: Spectral Ratios Confirm Mexico City Zoning. *Bulletin of the Seismological Society of America*, 108(6), 3289-3299. <https://doi.org/https://doi.org/10.1785/0120180100>
- Earthquake Engineering Field Investigation Team Japan. (2011). The Mw9.0 Tohoku Earthquake and Tsunami of 11th March 2011 Field Report [Technical report on the 2011 Tohoku Earthquake and Tsunami in Japan].
- Efraín Ovando-Shelley, A. O., & Santoyo, E. (2013). Effects of regional subsidence and earthquakes on architectural monuments in Mexico City. *Boletín de la Sociedad Geológica Mexicana*, 65(1), 123-142.
- Ferreira, A. J. M. (2008). *MATLAB Codes for Finite Element Analysis: Solids and Structures* (1.^a ed.). Springer Dordrecht. <https://doi.org/10.1007/978-1-4020-9200-8>

- Gazetas, G. (1991). Foundation Vibrations. En H.-Y. Fang (Ed.), *Foundation Engineering Handbook* (pp. 553-593). Springer US. https://doi.org/10.1007/978-1-4615-3928-5_15
- Ghavami, P. (2015). *Mechanics of Materials: An Introduction to Engineering Technology*. Springer Cham. <https://doi.org/https://doi.org/10.1007/978-3-319-07572-3>
- Gloyna, F. L. (1964). *Applications of the stiffness method for matrix structural analysis* [Tesis doctoral, Texas Tech University].
- Gobierno de la Ciudad de México. (2017). Normas Técnicas Complementarias para Diseño por Sismo [Gaceta Oficial de la Ciudad de México].
- Grupo de Trabajo de la Red Sísmica de la Ciudad de México. (2021). Red Sísmica de la Ciudad de México [Enero-marzo]. *Ciencia*, 72(1), 60-67.
- Hibbeler, R. C. (2022). *Análisis estructural*. Pearson Educación.
- Hori, M. (2011). *Introduction to Computational Earthquake Engineering* (2nd). Imperial College Press. <https://www.worldscientific.com/doi/abs/10.1142/q0132>
- Instituto de Geofísica, UNAM. (2017). Base de Datos de Registros Acelerográficos de la RAI-UNAM [Registro del sismo del 19 de septiembre de 2017 (M 7.1)]. Consultado el 10 de febrero de 2026, desde <https://aplicaciones.iingen.unam.mx/AcelerogramasRSM/>
- Kato, B., & Wang, G. (2022). Seismic site-city interaction analysis of super-tall buildings surrounding an underground station: a case study in Hong Kong. *Bulletin of Earthquake Engineering*, 20(3), 1431-1454. <https://doi.org/10.1007/s10518-021-01295-7>
- Khosravikia, F., Mahsuli, M., & Ghannad, M. A. (2018). Soil-Structure Interaction in Seismic Design Code: Risk-Based Evaluation. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 4(4), 04018033. <https://doi.org/10.1061/AJRUA6.0000984>
- Klügel, J.-U. (2008). Seismic Hazard Analysis — Quo vadis? *Earth-Science Reviews*, 88(1), 1-32. <https://doi.org/https://doi.org/10.1016/j.earscirev.2008.01.003>
- Kramer, S. (1996). *Geotechnical Earthquake Engineering*. Prentice Hall. <https://books.google.com.mx/books?id=sC22QgAACAAJ>

- Lagos, R., Lafontaine, M., Bonelli, P., Boroschek, R., Guendelman, T., Massone, L. M., Saragoni, R., Rojas, F., & Yañez, F. (2021). The quest for resilience: The Chilean practice of seismic design for reinforced concrete buildings. *Earthquake Spectra*, 37(1), 26-45. <https://doi.org/10.1177/8755293020970978>
- Leonardo Ramírez Guzmán, M. G. C. R. E. y. M. A. C. L. (2022). Interacción Sitio-Ciudad: el efecto de la ciudad en el movimiento del terreno en el Valle de México. *Instituto de Ingeniería, UNAM*.
- Lu, X., Tian, Y., Wang, G., & Huang, D. (2018). A numerical coupling scheme for nonlinear time history analysis of buildings on a regional scale considering site-city interaction effects. *Earthquake Engineering & Structural Dynamics*, 47(13), 2708-2725. <https://doi.org/https://doi.org/10.1002/eqe.3108>
- Mayoral, J., Asimaki, D., Tepalcapa, S., Wood, C., Roman-de la Sancha, A., Hutchinson, T., Franke, K., & Montalva, G. (2019). Site effects in Mexico City basin: Past and present. *Soil Dynamics and Earthquake Engineering*, 121, 369-382. <https://doi.org/https://doi.org/10.1016/j.soildyn.2019.02.028>
- Mayoral, J. M., & Pérez, M. (2024). Basin boundary seismic effects in Mexico City southern region. *Bulletin of Earthquake Engineering*, 22(3), 845-876. <https://doi.org/10.1007/s10518-023-01812-w>
- McGuire, W., Gallagher, R. H., & Ziemian, R. D. (2000). *Matrix structural analysis*. Bucknell University Faculty Books.
- Mistumasa Midorikawa, M. I., Izuru Okawa, & Teshigawara, M. (2003). Performance-Based Seismic Design Code for Buildings in Japan. *Earthquake Engineering and Engineering Seismology*, 4(1), 15-25.
- Oñate, E., & Rojek, J. (2004). Combination of Discrete Element and Finite Element Methods for Dynamic Analysis of Geomechanics Problems. *Computer Methods in Applied Mechanics and Engineering*, 193, 3087-3128. https://www.scipedia.com/public/Onate_Rojek_2004a
- Pailler, D., & Bard, E. (2002). (Table 1) Geochemical analytical data for sediment core MD95-2042 [In supplement to: Pailler, D; Bard, E (2002): High frequency palaeoceanographic changes during the past 140000 yr recorded by the organic matter in sediments of the Iberian Margin. *Palaeogeography, Palaeoclimatology, Palaeoecology*, 181(4), 431-452, [https://doi.org/10.1016/S0031-0182\(01\)00444-8](https://doi.org/10.1016/S0031-0182(01)00444-8)]. En *High frequency palaeoceanographic changes during the past 140000 yr recorded by the organic matter in sediments of the Iberian Margin*. PANGAEA. <https://doi.org/10.1594/PANGAEA.96864>
- Pinto-Vega, F. J., & Ledezma, C. (2019). Interacción suelo-estructura en edificios de gran altura con subterráneos en Santiago, Chile. *Obras y proyectos*, 66-75.

- Quinde, P., & Reinoso, E. (2020). Subsoil characteristics of Mexico City, acceleration and hysteretic energy spectra for the Mexico earthquake of September 19, 2017. *Geofísica Internacional*, 59(4), 234-256.
- Ramos Jiménez, A. S. (2008). *Revisión al Reglamento de Construcciones para el Distrito Federal: versión 1993* [Recuperado de Repositorio UNAM]. <https://repositorio.unam.mx/contenidos/372839>
- Sánchez-Sesma, F. J., & Luzón, F. (1996). Can horizontal P waves be trapped and resonate in a shallow sedimentary basin? *Geophysical Journal International*, 124(1), 209-214. <https://doi.org/10.1111/j.1365-246X.1996.tb06364.x>
- Sargiotis, D. (2025). *MATLAB for Civil Engineers: From Basics to Advanced Applications* (1.^a ed.). Springer Cham. <https://doi.org/10.1007/978-3-031-84673-1>
- Vicencio, F., Alexander, N. A., & Saavedra Flores, E. I. (2023). A State-of-the-Art review on Structure-Soil-Structure interaction (SSSI) and Site-City interactions (SCI). *Structures*, 56, 105002. <https://doi.org/https://doi.org/10.1016/j.istruc.2023.105002>
- Villafuerte Lujano, I. A., & Fernández-Dávila Gonzales, V. I. (2024). Seismic response of 8-story reinforced concrete buildings with irregular floor plan. *Ingeniería*, 34(2), 21-37.
- Wada, B. (1965). *Stiffness matrix structural analysis* (inf. téc.). National Aeronautics y Space Administration.
- Yigit Isbilibiroglu, R. T., & Bielak, J. (2015). Coupled Soil-Structure Interaction Effects of Building Clusters during Earthquakes. *Earthquake Spectra*, 31(1), 463-500. <https://doi.org/10.1193/102412EQS315M>



Resultados de MATLAB

A.1. RESULTADOS DE GENERACIÓN DE LA GEOMETRÍA DE LA ESTRUCTURA

```
>> marco3d_editable.m
Creando columnas...
   300 columnas creadas
Creando vigas en dirección Y...
   240 vigas en Y creadas
Creando vigas en dirección X...
   240 vigas en X creadas
RESUMEN DE ELEMENTOS
```

Columnas:	300
Vigas Y:	240
Vigas X:	240
TOTAL:	780 elementos

```
Clasificando elementos por tipo...
   300 columnas clasificadas
   480 vigas clasificadas
```

```
Graficando losas...
   192 losas graficadas
Asignando restricciones de frontera...
   25 nodos restringidos (base)
   300 nodos libres
```

TOOLTIPS INTERACTIVOS ACTIVADOS

- Haz clic en cualquier nodo o barra
 - Aparecerá información detallada
-
-

Validando consistencia del modelo...

- Nodos en xyz: 325 (coincide con nN)
- GDL totales: 1950 (6 por nodo)

Construyendo matriz de conectividad LaGe...

- LaGe: 780 × 2 (elementos × nodos/elemento)

Validando índices de nodos en LaGe...

- Índices de nodos válidos: 1 a 325
- Total de nodos disponibles: 325

Ensamblando matriz de masa...

Matriz inicializada: 1950 × 1950 (sparse)

Agregando masa de columnas...

Agregando masa de vigas...

- Masa de 480 vigas agregada
- Masa de 300 columnas agregada

Agregando inercia rotacional...

- Inercia rotacional agregada a 325 nodos

Agregando masa de losas...

Losa: espesor=25 cm, masa=600 kg/m²

- Masa de losas agregada a 300 nodos
- Masa total de losas: 3456000 kg (3456.0 ton)
- Verificación: error = 0.00% (esperado < 0.1%)

Calculando masa total por nodo...

- Masa nodal calculada
 - Mínima: 1050.0 kg
 - Máxima: 24060.0 kg
 - Promedio: 15496.6 kg
 - TOTAL: 5036400 kg (5036.4 ton)

Generando gráfica de distribución de masas...

- Gráfica generada

Implementando diafragmas rígidos...

- Nivel 1: Nodo master = 158, Slaves = 24
- Nivel 2: Nodo master = 159, Slaves = 24
- Nivel 3: Nodo master = 160, Slaves = 24
- Nivel 4: Nodo master = 161, Slaves = 24
- Nivel 5: Nodo master = 162, Slaves = 24

Nivel 6: Nodo master = 163, Slaves = 24
Nivel 7: Nodo master = 164, Slaves = 24
Nivel 8: Nodo master = 165, Slaves = 24
Nivel 9: Nodo master = 166, Slaves = 24
Nivel 10: Nodo master = 167, Slaves = 24
Nivel 11: Nodo master = 168, Slaves = 24
Nivel 12: Nodo master = 169, Slaves = 24
 12 diafragmas definidos

>>

A.2. RESULTADOS DE GENERACIÓN DE LA MATRIZ DE RIGIDEZ

>> mrigidez.m

ENSAMBLAJE DE MATRIZ DE RIGIDEZ GLOBAL

Información del modelo:

Nodos: 325
Barras: 780
GDL: 1950

Inicializando matrices...

Matriz K: 1950 × 1950 (sparse)

Ensamblando matriz de rigidez...

300 columnas ensambladas
 480 vigas ensambladas
 Total: 780 elementos

Verificando propiedades de la matriz K...

Simetría: error = 0.00e+00 (debe ser < 1e-10)
Elementos no cero: 18710 de 3802500 (99.5% sparse)
Diagonal: 1950/1950 elementos positivos

APLICANDO DIAFRAGMA RÍGIDO (Master-Slave)

Identificando GDL dependientes...

GDL dependientes: 864
GDL originales: 1950
Construyendo matriz de transformación...
 Matriz T: 1950 → 1086 GDL
Transformando matrices K y M...

- K: 1950 × 1950 → 1086 × 1086
- M: 1950 × 1950 → 1086 × 1086
- Reducción: 44.3% de GDL eliminados

Diafragmas aplicados exitosamente:

12 pisos con diafragma rígido
GDL eliminados: 864
GDL restantes: 1086

Aplicando condiciones de frontera...

GDL restringidos (empotramientos): 150
GDL libres: 936

Matrices reducidas generadas:

Tamaño: 936 × 936
K_red elementos no cero: 12171
M_red elementos no cero: 336

Verificando definición positiva de K_red...

Autovalor mínimo: 2.719709e+06
Autovalor máximo: 3.696099e+11
Número de condición: 1.36e+05

- K_red es definida positiva

Vector de fuerzas externas inicializado (ceros)

Tamaño: 1086 × 1

Generando gráficas de estructura de matrices...

- Gráficas generadas

RESUMEN DE MATRICES GENERADAS

Diafragma rígido:

Aplicado: SÍ
Pisos: 12
GDL eliminados: 864

Matriz de Rigidez:

Kglobal: 1086 × 1086 (con diafragma)
K_red: 936 × 936 (sin empotramientos)

Matriz de Masa:

Mglobal: 1086 × 1086 (con diafragma)
M_red: 936 × 936 (sin empotramientos)

Vector de Fuerzas:

F_ext: 1086 × 1 (inicializado en ceros)

GDL:

Total (con diafragma): 1086

Restringidos: 150

Libres: 936

Ensamblaje completado exitosamente

A.3. RESULTADOS DEL ANÁLISIS ESTÁTICO

>> análisis_estatico.m

ANÁLISIS ESTÁTICO - CARGA PUNTUAL

Información del modelo:

Nodos totales: 325

GDL totales: 1086

Niveles: 12

Nodo seleccionado para carga puntual:

Nodo: 325

Posición: X=20.0 m, Y=24.0 m, Z=42.0 m

(Esquina posterior-derecha de azotea)

Condiciones de frontera:

GDL restringidos: 150 (empotramiento base)

GDL libres: 936

Definiendo carga puntual...

Inicializando fuerzas en espacio completo (1950 GDL)

Transformando fuerzas al espacio con diafragma...

F_ext: 1950 → 1086 GDL

Fuerzas transformadas

Nodo cargado: 325

Ubicación: (20.0, 24.0, 42.0) m

Fuerza en X: 100 kN

Fuerza en Y: 0 kN

Fuerza en Z: 0 kN

Resolviendo sistema de ecuaciones...

Sistema resuelto

Método: Eliminación directa (backslash operator)

Tamaño del sistema: 936 × 936

Reconstruyendo desplazamientos completos...

Desplazamientos: 1086 → 1950 GDL

Analizando desplazamientos...

Desplazamiento máximo:

Magnitud: 0.015026 m = 15.03 mm

Nodo: 65

Ubicación: (0.0, 24.0, 42.0) m

Nodo cargado (325):

Ux = 0.013736 m = 13.736 mm

Uy = -0.006089 m = -6.089 mm

Uz = -0.000224 m = -0.224 mm

Desplazamientos por nivel:

Nivel	Altura	Ux (mm)	Uy (mm)	Uz (mm)
0	0.00	0.000	0.000	0.000
1	3.50	0.301	0.000	0.000
2	7.00	0.804	0.000	0.000
3	10.50	1.349	0.000	0.000
4	14.00	1.907	0.000	0.000
5	17.50	2.472	0.000	0.000
6	21.00	3.043	0.000	0.000
7	24.50	3.619	0.000	0.000
8	28.00	4.199	0.000	0.000
9	31.50	4.781	0.000	0.000
10	35.00	5.364	0.000	0.000
11	38.50	5.935	0.000	0.000
12	42.00	6.429	0.000	0.000

Desplazamientos del entrepiso:

Piso	Drift_X (%)	Drift_Y (%)	Drift_Z (%)
------	-------------	-------------	-------------

1	0.0086	0.0000	0.0000
2	0.0144	0.0000	0.0000
3	0.0156	0.0000	0.0000
4	0.0159	0.0000	0.0000
5	0.0162	0.0000	0.0000
6	0.0163	0.0000	0.0000
7	0.0165	0.0000	0.0000
8	0.0166	0.0000	0.0000
9	0.0166	0.0000	0.0000
10	0.0166	0.0000	0.0000
11	0.0163	0.0000	0.0000
12	0.0141	0.0000	0.0000

Calculando fuerzas internas en elementos...

780 elementos procesados

Generando visualización...

Visualización generada

RESUMEN DE ANÁLISIS ESTÁTICO

Carga aplicada:

Nodo: 325 (azotea)
 Fx: 100 kN
 Fy: 0 kN
 Fz: 0 kN

Desplazamiento máximo:

Magnitud: 15.026 mm
 Nodo: 65

Desplazamiento en nodo cargado:

Ux: 13.736 mm
 Uy: -6.089 mm
 Uz: -0.224 mm

Drift máximo:

Dirección X: 0.0166 % (piso 10)

Análisis estático completado

A.4. RESULTADOS DEL ANÁLISIS DINÁMICO

```
>> analisis_dinamico.m
  ANALISIS DINAMICO MODAL
Nodos: 325
GDL: 1086
```

```
Matriz de masa:
  Tamaño: 1086 x 1086
  Masa total: 5036 ton
```

```
Grados de libertad:
  Total: 1086
  Restringidos: 150
  Libres: 936
```

```
Matrices reducidas:
  K_red: 936 x 936
  M_red: 936 x 936
```

```
Resolviendo problema de autovalores...
12 modos calculados
```

FRECUENCIAS Y PERIODOS NATURALES

Modo	Frecuencia (Hz)	Periodo (s)
1	0.4123	2.4254
2	0.4357	2.2950
3	0.4906	2.0382
4	1.2644	0.7909
5	1.3345	0.7493
6	1.4981	0.6675

...

```
Normalizando modos respecto a masa...
Modos normalizados
```

```
Verificando ortogonalidad de modos...
```

```
Error ortogonalidad masa: 5.85e-14 (debe ser < 1e-10)
Error ortogonalidad rigidez: 8.29e-15 (debe ser < 1e-6)
```

```
Calculando factores de participacion...
Factores calculados
```

MASAS MODALES PARTICIPANTES

Modo	X (%)	Y (%)	Z (%)
1	0.00	80.52	0.00

2	80.52	0.00	0.00
3	0.00	0.00	0.00
4	0.00	9.77	0.00
5	9.98	0.00	0.00
6	0.00	0.00	0.00
Suma	90.50	90.29	0.00

Generando espectro de diseño (SASID)...

Espectro cargado: Q2.txt

Puntos: 51

Rango periodos: 0.01 - 5.00 s

Sa maximo: 0.32 g (@ Q=2)

Q base (SASID): 2.0

Q adicional: 1.0

Q efectivo: 2.0

Valores espectrales en modos principales:

Modo 1 (T=2.43s): Sa = 0.157 g

Modo 2 (T=2.30s): Sa = 0.181 g

Modo 3 (T=2.04s): Sa = 0.250 g

Modo 4 (T=0.79s): Sa = 0.316 g

Modo 5 (T=0.75s): Sa = 0.311 g

Modo 6 (T=0.67s): Sa = 0.300 g

Calculando desplazamientos espectrales...

Desplazamientos modales calculados

Aplicando combinacion SRSS...

Combinacion SRSS aplicada

Desplazamientos espectrales por nivel:

Nivel Altura Ux (cm) Uy (cm) Uz (cm)
 (Reconstruyendo desplazamientos con diafragma...)

Nivel	Altura	Ux (cm)	Uy (cm)
0	0.00	0.0000	0.0000
1	3.50	0.0000	2.0649
2	7.00	0.0000	5.5544
3	10.50	0.0000	9.2127
4	14.00	0.0000	12.7439
5	17.50	0.0000	16.0511
6	21.00	0.0000	19.0847
7	24.50	0.0000	21.8088
8	28.00	0.0000	24.1899
9	31.50	0.0000	26.1923

10	35.00	0.0000	27.7778
11	38.50	0.0000	28.9135
12	42.00	0.0000	29.6083

Generando grafica del espectro...

Grafica generada

Generando visualizacion de deformada...

Factor de amplificacion: 10.0

Deformacion maxima: 0.307 m

Visualizacion interactiva generada

Configurando analisis tiempo-historia...

Amortiguamiento: 5.0% critico

Modos a usar: 12

Cargando acelerograma...

Tipo: Acelerograma real (SCT2 - 19/09/2017)

(Convertido de Gal a m/s²)

Duracion: 951.0 s

Paso de tiempo: 0.0100 s

Puntos: 95100

Aceleracion maxima: 0.917 m/s² (0.09 g)

Generando grafica del acelerograma...

Grafica generada

Iniciando integracion temporal (Newmark)...

Parametros Newmark: $g=0.50$, $b^2=0.25$

Direccion de excitacion: Y

Integrando 12 modos...

Modo 3/12 completado

Modo 6/12 completado

Modo 9/12 completado

Modo 12/12 completado

Integracion completada

Reconstruyendo desplazamientos fisicos...

Desplazamientos reconstruidos

Tamaño: 1086 GDL x 95100 pasos de tiempo

Tamaño expandido: 1950 GDL x 95100 pasos de tiempo

Desplazamientos maximos:

Ux: 0.000 m = 0.0 cm

Uy: 0.155 m = 15.5 cm

Uz: 0.002 m = 0.2 cm

Desplazamientos maximos tiempo-historia por nivel:

Nivel	Altura	Ux_max (cm)	Uy_max (cm)
0	0.00	0.0000	0.0000
1	3.50	0.0000	1.0999
2	7.00	0.0000	2.9553
3	10.50	0.0000	4.8932
4	14.00	0.0000	6.7538
5	17.50	0.0000	8.4894
6	21.00	0.0000	10.0771
7	24.50	0.0000	11.5014
8	28.00	0.0000	12.7409
9	31.50	0.0000	13.7718
10	35.00	0.0000	14.5736
11	38.50	0.0000	15.1375
12	42.00	0.0000	15.4776

Generando graficas de respuesta temporal...

Nodo seleccionado: 325 (azotea)

Graficas generadas (X, Y, Z)

A.5. RESULTADOS DE ANÁLISIS DE INTERACCIÓN SUELO-ESTRUCTURA

>> analisis_ise_superposicion.m

ANALISIS ISE - Metodo de Superposicion Simplificado

PASO 0: Cargando modelo estructural...

Ejecutando scripts previos...

Creando columnas...

OK 300 columnas creadas

Creando vigas en direccion Y...

OK 240 vigas en Y creadas

Creando vigas en direccion X...

OK 240 vigas en X creadas

RESUMEN DE ELEMENTOS

=====

Columnas: 300
Vigas Y: 240
Vigas X: 240
TOTAL: 780 elementos

=====
Clasificando elementos por tipo...

OK 300 columnas clasificadas
OK 480 vigas clasificadas

Graficando losas...

OK 192 losas graficadas

Asignando restricciones de frontera...

OK 25 nodos restringidos (base)
OK 300 nodos libres

TOOLTIPS INTERACTIVOS ACTIVADOS

=====
-> Haz clic en cualquier nodo o barra
-> Aparecera informacion detallada
=====

Validando consistencia del modelo...

OK Nodos en xyz: 325 (coincide con nN)
OK GDL totales: 1950 (6 por nodo)

Construyendo matriz de conectividad LaGe...

OK LaGe: 780 x 2 (elementos x nodos/elemento)

Validando indices de nodos en LaGe...

OK Indices de nodos validos: 1 a 325
OK Total de nodos disponibles: 325

Ensamblando matriz de masa...

Matriz inicializada: 1950 x 1950 (sparse)

Agregando masa de columnas...

Agregando masa de vigas...

OK Masa de 480 vigas agregada

OK Masa de 300 columnas agregada

Agregando inercia rotacional...

OK Inercia rotacional agregada a 325 nodos

Agregando masa de losas...

Losa: espesor=25 cm, masa=600 kg/m²

OK Masa de losas agregada a 300 nodos

OK Masa total de losas: 3456000 kg (3456.0 ton)

OK Verificacion: error = 0.00% (esperado < 0.1%)

Calculando masa total por nodo...

OK Masa nodal calculada
Minima: 1050.0 kg
Maxima: 24060.0 kg
Promedio: 15496.6 kg
TOTAL: 5036400 kg (5036.4 ton)

Generando grafica de distribucion de masas...
OK Grafica generada

Implementando diafragmas rigidos...

Nivel 1: Nodo master = 158, Slaves = 24
Nivel 2: Nodo master = 159, Slaves = 24
Nivel 3: Nodo master = 160, Slaves = 24
Nivel 4: Nodo master = 161, Slaves = 24
Nivel 5: Nodo master = 162, Slaves = 24
Nivel 6: Nodo master = 163, Slaves = 24
Nivel 7: Nodo master = 164, Slaves = 24
Nivel 8: Nodo master = 165, Slaves = 24
Nivel 9: Nodo master = 166, Slaves = 24
Nivel 10: Nodo master = 167, Slaves = 24
Nivel 11: Nodo master = 168, Slaves = 24
Nivel 12: Nodo master = 169, Slaves = 24
OK 12 diafragmas definidos

ENSAMBLAJE DE MATRIZ DE RIGIDEZ GLOBAL

=====

Informacion del modelo:

Nodos: 325
Barras: 780
GDL: 1950

Inicializando matrices...

Matriz K: 1950 x 1950 (sparse)

Ensamblando matriz de rigidez...

OK 300 columnas ensambladas
OK 480 vigas ensambladas
OK Total: 780 elementos

Verificando propiedades de la matriz K...

Simetria: error = 0.00e+00 (debe ser < 1e-10)
Elementos no cero: 18710 de 3802500 (99.5% sparse)
Diagonal: 1950/1950 elementos positivos

APLICANDO DIAFRAGMA RIGIDO (Master-Slave)

=====

Identificando GDL dependientes...

GDL dependientes: 864

GDL originales: 1950

Construyendo matriz de transformacion...

OK Matriz T: 1950 -> 1086 GDL

Transformando matrices K y M...

OK K: 1950 x 1950 -> 1086 x 1086

OK M: 1950 x 1950 -> 1086 x 1086

OK Reduccion: 44.3% de GDL eliminados

Diafragmas aplicados exitosamente:

12 pisos con diafragma rigido

GDL eliminados: 864

GDL restantes: 1086

Aplicando condiciones de frontera...

GDL restringidos (empotramientos): 150

GDL libres: 936

Matrices reducidas generadas:

Tamano: 936 x 936

K_red elementos no cero: 12171

M_red elementos no cero: 336

Verificando definicion positiva de K_red...

Autovalor minimo: 2.719709e+06

Autovalor maximo: 3.696099e+11

Numero de condicion: 1.36e+05

OK K_red es definida positiva

Vector de fuerzas externas inicializado (ceros)

Tamano: 1086 x 1

Generando graficas de estructura de matrices...

OK Graficas generadas

RESUMEN DE MATRICES GENERADAS

=====

Diafragma rigido:

Aplicado: SI

Pisos: 12

GDL eliminados: 864

Matriz de Rigidez:

K_{global}: 1086 x 1086 (con diafragma)

K_{red}: 936 x 936 (sin empotramientos)

Matriz de Masa:

M_{global}: 1086 x 1086 (con diafragma)

M_{red}: 936 x 936 (sin empotramientos)

Vector de Fuerzas:

F_{ext}: 1086 x 1 (inicializado en ceros)

GDL:

Total (con diafragma): 1086

Restringidos: 150

Libres: 936

=====

OK Ensamblaje completado exitosamente

OK Modelo cargado

- Nodos: 325

- GDLs libres: 936

PASO 1: Parametros Zona III

Vs=80 m/s, rho=1250 kg/m³, nu=0.499, G=8000 kPa

Ref: NTC-Cimentaciones (2017)

PASO 2: Cimentacion

Losa: 24.0x20.0 m, r0=12.36 m

PASO 3: Identificar nodos base

25 nodos en z=0

PASO 4: Analisis modal BASE RIGIDA

Primeros 3 modos:

Modo 1: T=2.425 s, f=0.412 Hz

Modo 2: T=2.295 s, f=0.436 Hz

Modo 3: T=2.038 s, f=0.491 Hz

PASO 5: Geometria de cimentacion e impedancias estaticas

Ref: Gazetas (1991); Aviles Lopez et al. (2016), Sec. 6.3.2
Formulacion: disco circular rigido sobre semiespacio elastico
(Luco & Westmann 1971, extendida por Gazetas 1983, 1991)

NOTA: Los coeficientes dinamicos (Ch, Cr) dependen de la frecuencia efectiva T_e y se calculan en el PASO 6 mediante iteracion (NTC-Sismo 2020, Sec. 8.2.3).

Radios equivalentes:

r_0 (traslacion): 12.361 m [= $\sqrt{A/\pi}$]
 r_{0r} (rotacion X): 11.947 m [= $(4I_x/\pi)^{1/4}$]
 r_{0r} (rotacion Y): 13.087 m [= $(4I_y/\pi)^{1/4}$]
 r_{0r} (promedio): 12.517 m

Rigideces estaticas:

$K_h = 8 \cdot G \cdot r_0 / (2 - \nu) = 5.2704e+08$ N/m
 $K_r = 8 \cdot G \cdot r_{0r}^3 / [3(1 - \nu)] = 8.3509e+10$ N*m/rad

Coeficientes de amortiguamiento (evaluados en w y -- ver PASO 6):

$ch = 0.60$ (traslacion horizontal, Gazetas 1991)
 $cr = 0.40$ (cabeceo, Gazetas 1991)

Verificacion amortiguamiento por radiacion:

Frecuencia resonante del estrato: $f_s = 0.4000$ Hz
Frecuencia fundamental estructura: $f_1 = 0.4123$ Hz
OK: $f_1 > f_s \rightarrow$ amortiguamiento radiativo activo.

PASO 6: Analisis modal BASE FLEXIBLE

METODO: Periodo efectivo por suma de cuadrados (NTC-Sismo 2020)

Ref principal: NTC-Sismo (2020), ecs. 8.2.7-8.2.12

Ref secundaria: Aviles Lopez et al. (2016), Sec. A4.3

NOTA: Modo traslacional dominante en X es el modo 2 ($T=2.295$ s).
El modo 1 es torsional. Se usa modo 2 para el ISE.

Masa efectiva modal (modo 1): $M_e = 4034.0$ ton
Peso efectivo modal: $W_e = 39574.0$ kN
Masa total traslacional: $M_t = 5010.1$ ton
Razon $M_e/M_t = 0.805$ (rango tipico 0.60-0.85)

Altura efectiva (modal): $H_e = 28.47$ m
Altura efectiva (aprox 0.7H): $H_e = 29.40$ m <- referencia NTC estatico

Se usara He modal para mayor rigor.

ITERACION -- Impedancias evaluadas en $w_{e_tilde} = 2\pi/T_{e_tilde}$

Iter	T_{e_tilde} (s)	Th (s)	Tr (s)	Error
1	2.667497	0.549702	1.243469	1.62e-01
2	2.667497	0.549702	1.243469	0.00e+00

OK Convergencia en 2 iteracion(es) (tol = 1e-06)

Periodo efectivo -- Ec. 8.2.7 NTC-Sismo

$T_{e_tilde}^2 = T_e^2 + T_h^2 + T_r^2$ (solucion iterada)

Te (base rigida): 2.2950 s
Th (traslacion): 0.5497 s
Tr (cabeceo): 1.2435 s
 T_{e_tilde} (base flexible): 2.6675 s
DT = +0.3725 s (DT/Te = +16.2%)

Contribucion al incremento $T_{e_tilde}^2 - T_e^2$:

Traslacion (T_h^2): 16.3%
Cabeceo (T_r^2): 83.7%

Rango esperado Zona III: +20% a +55%

Refs: Aviles et al. (2016); Mayoral et al. (2019);
Quinde & Reinoso (2020)

*** NOTA: DT < 20%. Verificar parametros del suelo.

Amortiguamiento efectivo (w_{e_tilde} convergida):

de (estructura): 0.0500 (5.0%)
dh (traslacion): 0.0143 (1.4%)
dr (cabeceo): 0.0098 (1.0%)
 d_{e_tilde} (sistema): 0.0500 (5.0%)
Ch (convergado): 1.7782e+07 N*s/m
Cr (convergado): 1.9262e+09 N*m*s/rad

Comparacion de periodos (modo 1 modificado por ISE):

Modo	T_{rigida}	$T_{flexible}$	DT	DT%	
1	2.4254	2.4254	+0.0000	+0.0%	<- ISE
2	2.2950	2.6675	0.000	0.0%	
3	2.0382	2.0382	0.000	0.0%	

NOTA: Los modos superiores no se modifican por ISE.

Ref: NTC-Sismo (2020), Sec. 8.2; Aviles Lopez et al. (2016), Sec. 6.5.3

PASO 7: Acelerograma 19S-2017

Duracion: 951.0 s, a_max=0.92 m/s²

PASO 8: Analisis tiempo-historia (base rigida)

Integrando 95100 pasos con Newmark-beta (aceleracion promedio)...

Excitacion: X e Y simultaneos (mismo acelerograma, en fase)

Regularizacion M_red: 600 GDLs con masa=0, masa ficticia=7.53e-03 kg

Amortiguamiento Rayleigh: alpha=1.4076e-01, beta=1.7626e-02

(Calibrado para xi=5% en modos 1 y 3)

GDLs en vector iota_x (Ux activos): 12

GDLs en vector iota_y (Uy activos): 12

GDLs restantes (rotaciones, Uz): no excitados

Factorizando K_eff (LU)...

OK Factorizacion completada

Integrando 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% OK (33.7 s)

u_max horizontal (envolvente X-Y) = 0.1515 m (15.15 cm)

PASO 9: Graficas

OK periodos.png

ANALISIS COMPLETADO

B

Códigos de Análisis [MATLAB R2024b]

B.1. marco3d_editable.m

```
1 %% marco3d_editable
2 %% 0. Flag de configuración
3 % USE_DIAPHRAGM = true -> condensación master-slave por piso (modelo original)
4 % USE_DIAPHRAGM = false -> nodos independientes, shells de membrana activos
5 %
6 % NOTA TÉCNICA: Con USE_DIAPHRAGM = false, la rigidez de membrana de las
7 % losas contribuye efectivamente a K_red porque los nodos de cada panel
8 % son cinemáticamente independientes y el panel puede deformarse en su
9 % plano. Con USE_DIAPHRAGM = true, la condensación master-slave impone
10 % desplazamiento lateral uniforme en cada piso, produciendo deformación
11 % de membrana nula en todos los paneles y anulando su contribución a K_red.
12 % Esta incompatibilidad entre diafragma rígido y rigidez de membrana de
13 % losas es la motivación para la presente decisión de modelado.
14 % Ref: Cook, Malkus, Plesha & Witt (2002), Cap. 6; Chopra (2012), Cap. 13.
15 USE_DIAPHRAGM = true;
16
17 %% 1. Parámetros
18 nNiv = 12; % Niveles (pisos)
19 nCruj = 4; % Crujías en Y
20 nProf = 4; % Crujías en X
21
22 L = 6; % [m] Distancia entre ejes en Y
23 h = 3.5; % [m] Altura de cada piso
24 w = 5; % [m] Distancia entre ejes en X
25
26 nN = (nNiv+1)*(nCruj+1)*(nProf+1); % Total de nodos
27 ngdl = 6 * nN; % Número total de grados de libertad (6 por nodo)
28
29 altura_total = nNiv * h; % Altura total del edificio (de PB a azotea)
```

```

30
31 % 1.1 Definición de los parámetros del material y geometría de las
32 % secciones
33 E = 2.1e10; % Módulo de elasticidad del concreto armado (Pa = N/m^2)
34 G = E/(2*(1+0.3)); % asumiendo Poisson = 0.3
35
36 % 1.2 COLUMNAS
37 b_col = 0.50; % [m] Base
38 h_col = 0.50; % [m] Altura
39 A_col = b_col * h_col;
40 Iy_col = (b_col * h_col^3) / 12;
41 Iz_col = (h_col * b_col^3) / 12;
42
43
44 % 1.3 Constante de torsión J (método híbrido)
45 ratio_col = b_col / h_col;
46 tol = 0.01; % Tolerancia para considerar "cuadrada"
47
48 if abs(ratio_col - 1.0) < tol
49     % 1.3.1 Sección CUADRADA: usar valor exacto
50     beta_col = 0.141;
51     J_col = beta_col * b_col * h_col^3;
52 else
53     % 1.3.2 Sección RECTANGULAR: usar fórmula aproximada
54     if b_col >= h_col
55         beta_col = (1/3) * (1 - 0.63 * (h_col / b_col));
56         J_col = beta_col * b_col * h_col^3;
57     else
58         beta_col = (1/3) * (1 - 0.63 * (b_col / h_col));
59         J_col = beta_col * h_col * b_col^3;
60     end
61 end
62
63 % 1.4 VIGAS
64 b_vig = 0.30; % [m] Base
65 h_vig = 0.50; % [m] Peralte
66 A_vig = b_vig * h_vig;
67 Iy_vig = (b_vig * h_vig^3) / 12;
68 Iz_vig = (h_vig * b_vig^3) / 12;
69
70 % 1.4.1 Constante de torsión J para vigas
71 ratio_vig = b_vig / h_vig; % = 0.30/0.50 = 0.6
72 tol = 0.01;
73
74 if abs(ratio_vig - 1.0) < tol
75     % Sección cuadrada
76     beta_vig = 0.141;
77     J_vig = beta_vig * b_vig * h_vig^3;
78 else
79     % Sección rectangular
80     if b_vig >= h_vig
81         beta_vig = (1/3) * (1 - 0.63 * (h_vig / b_vig));
82         J_vig = beta_vig * b_vig * h_vig^3;
83     else
84         % b_vig < h_vig

```

```

85     beta_vig = (1/3) * (1 - 0.63 * (b_vig / h_vig));
86     J_vig = beta_vig * h_vig * b_vig^3;
87     end
88 end
89
90
91 %% 2. Generar nodos
92 N = struct;
93 iCont = 1;
94 for iP = 1:(nProf+1)
95     for iC = 1:(nCruj+1)
96         for iNiv_i = 1:(nNiv+1)
97             N(iCont).x = (iP-1) * w;
98             N(iCont).y = (iC-1) * L;
99             N(iCont).z = (iNiv_i-1) * h;
100            N(iCont).num = iCont; % Etiqueta
101            iCont = iCont + 1;
102        end
103    end
104 end
105
106 % 2.1 Construcción de la matriz xyz (coordenadas nodales)
107 % Esta matriz se usa para detectar nodos base, visualizar deformadas y aplicar
    ISS
108 xyz = zeros(nN, 3);
109 for i = 1:nN
110     xyz(i,1) = N(i).x;
111     xyz(i,2) = N(i).y;
112     xyz(i,3) = N(i).z;
113 end
114
115 % 2.1.1 Verificación de consistencia estructural
116 ngdl_esperado = 6 * size(xyz,1);
117 if ngdl ~= ngdl_esperado
118     error('[ERROR] ngdl = %d, pero se esperaban %d = 6 - nNodos. Verificar.',
119         ngdl, ngdl_esperado);
120 end
121
122 % 2.2 Inicialización de la estructura de barras
123 Barras = struct('nodo1',{},'nodo2',{});
124 kBarra = 0;
125
126 %% 3. Figura única para nodos y barras
127 figure('Name','Marco 3D','NumberTitle','off');
128 hold on; grid on; axis equal;
129 view(40,20);
130 xlabel('Eje X'); ylabel('Eje Y'); zlabel('Eje Z');
131 title('Marco 3D');
132
133 % 3.1 Graficar nodos (triángulo en z=0, bolita en z>0)
134 offset = min(L, w) * 0.05;
135
136 for iNod=1:nN
137     if N(iNod).z == 0

```

```

138     % Triángulo en la base
139     hNodo = plot3(N(iNod).x, N(iNod).y, N(iNod).z, '^', ...
140         'MarkerFaceColor','k','MarkerEdgeColor','k');
141     else
142         % Círculo en niveles superiores
143         hNodo = plot3(N(iNod).x, N(iNod).y, N(iNod).z, 'o', ...
144             'MarkerFaceColor','k','MarkerEdgeColor','k');
145     end
146     % Guarda info en UserData (para el data tip)
147     hNodo.UserData = N(iNod);
148
149     % Etiqueta de texto en azul
150     text(N(iNod).x+0.1, N(iNod).y+0.1, N(iNod).z+0.1, ...
151         num2str(N(iNod).num), 'Color','b','FontWeight','bold');
152 end
153
154 %% 4. Graficar barras
155 % CREAR COLUMNAS (elementos verticales)
156
157 fprintf('Creando columnas...\n');
158
159 for iP = 1:(nProf + 1)
160     for iC = 1:(nCruj + 1)
161         for iNiv = 1:nNiv
162             % Nodos extremos
163             idx1 = nodeIndex(iNiv, iC, iP, nNiv, nCruj);
164             idx2 = nodeIndex(iNiv + 1, iC, iP, nNiv, nCruj);
165
166             % Graficar columna
167             hBarra = plot3([N(idx1).x, N(idx2).x], ...
168                 [N(idx1).y, N(idx2).y], ...
169                 [N(idx1).z, N(idx2).z], ...
170                 'LineWidth', 2.5, ...
171                 'Color', [0.8, 0.2, 0.2]); % Rojo para columnas
172
173             % Guardar info mínima para tooltip
174             hBarra.UserData = struct('nodo1', idx1, 'nodo2', idx2, ...
175                 'tipo', 'columna');
176
177             % Calcular longitud
178             dx = N(idx2).x - N(idx1).x;
179             dy = N(idx2).y - N(idx1).y;
180             dz = N(idx2).z - N(idx1).z;
181             L_barra = sqrt(dx^2 + dy^2 + dz^2);
182
183             % Agregar a estructura Barras
184             kBarra = kBarra + 1;
185             Barras(kBarra).nodo1 = idx1;
186             Barras(kBarra).nodo2 = idx2;
187             Barras(kBarra).tipo = 'columna';
188             Barras(kBarra).L = L_barra;
189         end
190     end
191 end
192

```

```

193 fprintf(' [OK] %d columnas creadas\n', kBarra);
194
195 % 4.2 Vigas en Y
196 % CREAR VIGAS EN DIRECCIÓN Y
197
198
199 fprintf('Creando vigas en dirección Y...\n');
200 n_vigas_Y_inicial = kBarra;
201
202 for iP = 1:(nProf + 1)
203     for iNiv = 1:(nNiv + 1)
204         for iC = 1:nCruj
205             % Nodos extremos
206             idx1 = nodeIndex(iNiv, iC, iP, nNiv, nCruj);
207             idx2 = nodeIndex(iNiv, iC + 1, iP, nNiv, nCruj);
208
209             % Verificar que NO estén ambos en la base
210             % (no hay vigas conectando cimentación)
211             if N(idx1).z > 0 || N(idx2).z > 0
212                 % Graficar viga
213                 hBarra = plot3([N(idx1).x, N(idx2).x], ...
214                               [N(idx1).y, N(idx2).y], ...
215                               [N(idx1).z, N(idx2).z], ...
216                               'LineWidth', 2.0, ...
217                               'Color', [0.2, 0.6, 0.2]); % Verde para vigas Y
218
219                 % Info para tooltip
220                 hBarra.UserData = struct('nodo1', idx1, 'nodo2', idx2, ...
221                                         'tipo', 'viga_Y');
222
223                 % Calcular longitud
224                 dx = N(idx2).x - N(idx1).x;
225                 dy = N(idx2).y - N(idx1).y;
226                 dz = N(idx2).z - N(idx1).z;
227                 L_barra = sqrt(dx^2 + dy^2 + dz^2);
228
229                 % Agregar a estructura
230                 kBarra = kBarra + 1;
231                 Barras(kBarra).nodo1 = idx1;
232                 Barras(kBarra).nodo2 = idx2;
233                 Barras(kBarra).tipo = 'viga_Y';
234                 Barras(kBarra).L = L_barra;
235             end
236         end
237     end
238 end
239
240 n_vigas_Y = kBarra - n_vigas_Y_inicial;
241 fprintf(' [OK] %d vigas en Y creadas\n', n_vigas_Y);
242
243 % 4.3 Vigas en X
244 % CREAR VIGAS EN DIRECCIÓN X
245
246 fprintf('Creando vigas en dirección X...\n');
247 n_vigas_X_inicial = kBarra;

```

```

248
249 for iC = 1:(nCruj + 1)
250     for iNiv = 1:(nNiv + 1)
251         for iP = 1:nProf
252             % Nodos extremos
253             idx1 = nodeIndex(iNiv, iC, iP, nNiv, nCruj);
254             idx2 = nodeIndex(iNiv, iC, iP + 1, nNiv, nCruj);
255
256             % Verificar que NO estén ambos en la base
257             if N(idx1).z > 0 || N(idx2).z > 0
258                 % Graficar viga
259                 hBarra = plot3([N(idx1).x, N(idx2).x], ...
260                             [N(idx1).y, N(idx2).y], ...
261                             [N(idx1).z, N(idx2).z], ...
262                             'LineWidth', 2.0, ...
263                             'Color', [0.2, 0.2, 0.8]); % Azul para vigas X
264
265                 % Info para tooltip
266                 hBarra.UserData = struct('nodo1', idx1, 'nodo2', idx2, ...
267                                         'tipo', 'viga_X');
268
269                 % Calcular longitud
270                 dx = N(idx2).x - N(idx1).x;
271                 dy = N(idx2).y - N(idx1).y;
272                 dz = N(idx2).z - N(idx1).z;
273                 L_barra = sqrt(dx^2 + dy^2 + dz^2);
274
275                 % Agregar a estructura
276                 kBarra = kBarra + 1;
277                 Barras(kBarra).nodo1 = idx1;
278                 Barras(kBarra).nodo2 = idx2;
279                 Barras(kBarra).tipo = 'viga_X';
280                 Barras(kBarra).L = L_barra;
281             end
282         end
283     end
284 end
285
286 n_vigas_X = kBarra - n_vigas_X_inicial;
287 fprintf(' [OK] %d vigas en X creadas\n', n_vigas_X);
288
289 hold off;
290
291
292 fprintf(' RESUMEN DE ELEMENTOS\n');
293 fprintf('-----\n');
294 fprintf(' Columnas: %d\n', n_vigas_Y_inicial);
295 fprintf(' Vigas Y: %d\n', n_vigas_Y);
296 fprintf(' Vigas X: %d\n', n_vigas_X);
297 fprintf(' TOTAL: %d elementos\n', length(Barras));
298 fprintf('-----\n');
299
300 % SECCIÓN 4.4

```

```

301 % CLASIFICACIÓN DE ELEMENTOS POR TIPO
302 % Crea listas E_col y E_vig para cálculo de masa
303
304 fprintf('Clasificando elementos por tipo...\n');
305
306 E_col = [];
307 E_vig = [];
308
309 for e = 1:length(Barras)
310     i = Barras(e).nodo1;
311     j = Barras(e).nodo2;
312
313     dx = abs(N(j).x - N(i).x);
314     dy = abs(N(j).y - N(i).y);
315     dz = abs(N(j).z - N(i).z);
316
317     if dz > 0 && dx == 0 && dy == 0
318         % Es columna
319         E_col = [E_col; e, i, j];
320     elseif dz == 0
321         % Es viga
322         E_vig = [E_vig; e, i, j];
323     end
324 end
325
326 fprintf(' [OK] %d columnas clasificadas\n', size(E_col, 1));
327 fprintf(' [OK] %d vigas clasificadas\n', size(E_vig, 1));
328
329 total_clasificados = size(E_col, 1) + size(E_vig, 1);
330 if total_clasificados ~= length(Barras)
331     warning('Hay %d elementos sin clasificar', length(Barras) -
332         total_clasificados);
333 end
334
335 fprintf('\n');
336
337 % 4.4.1 GRAFICAR LOSAS (rectángulos semitransparentes)
338 fprintf('Graficando losas...\n');
339
340 % Recorrer cada nivel (excepto la base z=0)
341 for iNiv = 2:(nNiv + 1) % Desde nivel 1 hasta azotea
342
343     % Altura del nivel actual
344     z_nivel = (iNiv - 1) * h;
345
346     % Recorrer cada celda en planta
347     for iP = 1:nProf
348         for iC = 1:nCruj
349
350             % Coordenadas de las 4 esquinas de la losa (celda)
351             x_min = (iP - 1) * w;
352             x_max = iP * w;
353             y_min = (iC - 1) * L;
354             y_max = iC * L;

```

```

355     % Definir las 4 esquinas del rectángulo (en sentido antihorario)
356     X_losa = [x_min, x_max, x_max, x_min];
357     Y_losa = [y_min, y_min, y_max, y_max];
358     Z_losa = [z_nivel, z_nivel, z_nivel, z_nivel];
359
360     % Graficar losa como superficie (patch)
361     patch(X_losa, Y_losa, Z_losa, ...
362           [0.60, 0.60, 0.56], ... % Color rojo
363           'FaceAlpha', 0.8, ... % Transparencia 30%
364           'EdgeColor', [0.6, 0.6, 0.6], ... % Borde gris oscuro
365           'LineWidth', 0.5, ...
366           'LineStyle', '-');
367     end
368   end
369 end
370
371 fprintf(' [OK] %d losas graficadas\n', nProf * nCruj * nNiv);
372
373
374 % 4.5 Asignación de grados de libertad y restricciones de frontera
375 % Marca nodos empotrados (z=0) vs libres (z>0)
376 % CONVENCIÓN:
377 % restringido = 1 (empotrado, no se mueve)
378 % libre = 0 (puede moverse)
379
380 fprintf('Asignando restricciones de frontera...\n');
381
382 for iC = 1:nN
383   if N(iC).z == 0
384     % Nodo en la base: EMPOTRAMIENTO (todos los GDL restringidos)
385     N(iC).Ux = 1; % Restringido en X
386     N(iC).Uy = 1; % Restringido en Y
387     N(iC).Uz = 1; % Restringido en Z
388     N(iC).Rx = 1; % Rotación respecto a X restringida
389     N(iC).Ry = 1; % Rotación respecto a Y restringida
390     N(iC).Rz = 1; % Rotación respecto a Z restringida
391   else
392     % Nodo libre: puede moverse en todas direcciones
393     N(iC).Ux = 0; % Libre en X
394     N(iC).Uy = 0; % Libre en Y
395     N(iC).Uz = 0; % Libre en Z
396     N(iC).Rx = 0; % Libre para rotar respecto a X
397     N(iC).Ry = 0; % Libre para rotar respecto a Y
398     N(iC).Rz = 0; % Libre para rotar respecto a Z
399   end
400 end
401
402 % 4.5.1 Contar nodos restringidos
403 n_restringidos = sum([N.z] == 0);
404 fprintf(' [OK] %d nodos restringidos (base)\n', n_restringidos);
405 fprintf(' [OK] %d nodos libres\n\n', nN - n_restringidos);
406
407
408 % 4.6 Función nodeIndex
409 % Calcula índice lineal de nodo según su posición (i,j,k) en malla 3D

```

```

410
411 function idx = nodeIndex(iN, iC, iP, nNiv, nCruj)
412     % Desplazamiento por profundidad (bloques completos de X)
413     offset_profundidad = (iP - 1) * (nCruj + 1) * (nNiv + 1);
414
415     % Desplazamiento por crujía (columnas completas en Y)
416     offset_crujia = (iC - 1) * (nNiv + 1);
417
418     % Posición en nivel (Z)
419     offset_nivel = iN;
420
421     % Índice final
422     idx = offset_profundidad + offset_crujia + offset_nivel;
423 end
424
425 % 4.7 Activar datacursormode con UpdateFcn
426 % ACTIVAR TOOLTIPS INTERACTIVOS
427
428 dcm = datacursormode(gcf);
429 set(dcm, 'UpdateFcn', @customDataTip);
430
431 fprintf(' TOOLTIPS INTERACTIVOS ACTIVADOS\n');
432 fprintf('-----\n');
433 fprintf(' * Haz clic en cualquier nodo o barra\n');
434 fprintf(' * Aparecerá información detallada\n');
435 fprintf('-----\n');
436
437
438 % 4.8 Función para data tip
439 % FUNCIÓN: customDataTip
440 % Genera texto personalizado para tooltips de nodos y barras
441 % Callback para datacursormode que muestra información contextual
442 % al hacer clic en elementos gráficos (nodos o barras).
443 %
444 % INPUTS:
445 % ~ - Objeto (no usado, requerido por MATLAB)
446 % event_obj - Evento del datacursor con información del target
447 %
448 % OUTPUT:
449 % output_txt - Celda de strings para mostrar en el tooltip
450 %
451 % CASOS MANEJADOS:
452 % 1. Nodos: Muestra número y coordenadas
453 % 2. Barras: Muestra conectividad, tipo y longitud
454 % 3. Otro: Muestra solo posición
455
456 function output_txt = customDataTip(~, event_obj)
457     % Obtener objeto clickeado y su posición
458     target = get(event_obj, 'Target');
459     pos = get(event_obj, 'Position');
460
461     % Verificar si hay UserData
462     if isempty(target.UserData)

```

```

463     % Sin datos: mostrar solo posición
464     output_txt = {
465         sprintf('X: %.2f m', pos(1)), ...
466         sprintf('Y: %.2f m', pos(2)), ...
467         sprintf('Z: %.2f m', pos(3))
468     };
469     return;
470 end
471
472 % CASO 1: Es un nodo (tiene campo 'num')
473 if isfield(target.UserData, 'num')
474     nodo = target.UserData;
475     output_txt = {
476         sprintf('----- NODO %d -----',
477             nodo.num), ...
478         sprintf('X: %.2f m', nodo.x), ...
479         sprintf('Y: %.2f m', nodo.y), ...
480         sprintf('Z: %.2f m', nodo.z)
481     };
482     return;
483 end
484
485 % CASO 2: Es una barra (tiene campo 'nodol')
486 if isfield(target.UserData, 'nodol')
487     barra = target.UserData;
488
489     % Línea de encabezado
490     output_txt = {sprintf('----- BARRA
491         -----')};
492
493     % Conectividad (siempre presente)
494     output_txt{end+1} = sprintf('Nodo %d - Nodo %d', ...
495         barra.nodol, barra.nodo2);
496
497     % Tipo (si existe)
498     if isfield(barra, 'tipo')
499         output_txt{end+1} = sprintf('Tipo: %s', barra.tipo);
500     end
501
502     % Longitud (si existe)
503     if isfield(barra, 'L')
504         output_txt{end+1} = sprintf('Longitud: %.3f m', barra.L);
505     end
506
507     % Coordenadas punto i (si existen)
508     if isfield(barra, 'i')
509         output_txt{end+1} = sprintf('i: [%.2f, %.2f, %.2f] m', ...
510             barra.i(1), barra.i(2), barra.i(3));
511     end
512
513     % Coordenadas punto j (si existen)
514     if isfield(barra, 'j')
515         output_txt{end+1} = sprintf('j: [%.2f, %.2f, %.2f] m', ...
516             barra.j(1), barra.j(2), barra.j(3));
517     end
518 end

```

```

516         return;
517     end
518
519     % CASO 3: Otro tipo de objeto (fallback)
520     output_txt = {
521         'Objeto desconocido', ...
522         sprintf('Posición: [%.2f, %.2f, %.2f]', pos(1), pos(2), pos(3))
523     };
524 end
525
526
527
528 %% 5. Asignación de variables globales y Validación de Consistencia Estructural
529
530 fprintf('Validando consistencia del modelo...\n');
531
532 % 5.1 Verificar que número de nodos en xyz coincide con nN
533 n_nodos_xyz = size(xyz, 1);
534 if n_nodos_xyz ~= nN
535     error(['ERROR] xyz tiene %d nodos, pero nN = %d. Inconsistencia detectada.',
536         ...
537         n_nodos_xyz, nN]);
538 end
539 fprintf(' [OK] Nodos en xyz: %d (coincide con nN)\n', n_nodos_xyz);
540
541 % 5.2 Verificar que ngdl es consistente
542 ngdl_calculado = 6 * nN;
543 if ngdl ~= ngdl_calculado
544     error(['ERROR] ngdl = %d, pero debería ser %d = 6 - %d. Verifica definición
545         inicial.', ...
546         ngdl, ngdl_calculado, nN]);
547 end
548 fprintf(' [OK] GDL totales: %d (6 por nodo)\n', ngdl);
549
550 % 5.3 Construcción explícita de LaGe (Matriz de Conectividad) a partir de Barras
551 % LaGe(e,:) = [nodo_inicial, nodo_final] del elemento e
552 % Formato alternativo a Barras para compatibilidad con otros códigos
553
554 fprintf('Construyendo matriz de conectividad LaGe...\n');
555
556 n_elementos = length(Barras);
557 LaGe = zeros(n_elementos, 2);
558
559 for k = 1:n_elementos
560     LaGe(k, 1) = Barras(k).nodo1;
561     LaGe(k, 2) = Barras(k).nodo2;
562 end
563
564 fprintf(' [OK] LaGe: %d [OK]-- 2 (elementos [OK]-- nodos/elemento)\n',
565     n_elementos);
566
567 % 5.4 Validación de LaGe contra nodos reales (Validación de Índices Nodales
568 fprintf('Validando índices de nodos en LaGe...\n');

```

```

568
569 % 5.4.1 Verificar índice máximo
570 max_node_index = max(LaGe(:));
571 if max_node_index > size(xyz, 1)
572     error(['ERROR] Elemento con nodo %d, pero solo hay %d nodos en xyz.', ...
573           max_node_index, size(xyz, 1)];
574 end
575
576 % 5.4.2 Verificar índice mínimo
577 min_node_index = min(LaGe(:));
578 if min_node_index < 1
579     error(['ERROR] Elemento con nodo %d (< 1). Índices deben ser >= 1.', ...
580           min_node_index);
581 end
582
583 fprintf(' [OK] Índices de nodos válidos: %d a %d\n', min_node_index,
584         max_node_index);
585 fprintf(' [OK] Total de nodos disponibles: %d\n\n', size(xyz, 1));
586
587 %% 6. Cálculo de matriz de masa concentrada (parametrizada)
588 fprintf('Ensamblando matriz de masa...\n');
589
590 % 6.1 Parámetros de material (ya definido antes, pero se reitera por claridad)
591 rho = 2400; % [kg/m^3] Densidad del concreto armado
592
593 % 6.2 Inicializar matriz de masa global (sparse para eficiencia)
594 Mglobal = sparse(ngdl, ngdl);
595
596 fprintf(' Matriz inicializada: %d - %d (sparse)\n', ngdl, ngdl);
597
598 % 6.3 MASA DE COLUMNAS
599 fprintf(' Agregando masa de columnas...\n');
600
601 % 6.4 MASA DE VIGAS
602
603 fprintf(' Agregando masa de vigas...\n');
604
605 for e = 1:size(E_vig, 1)
606     % 6.4.1 Nodos extremos
607     ni = E_vig(e, 2);
608     nj = E_vig(e, 3);
609
610     % 6.4.2 Longitud de viga
611     dx = N(nj).x - N(ni).x;
612     dy = N(nj).y - N(ni).y;
613     dz = N(nj).z - N(ni).z;
614     Lv = sqrt(dx^2 + dy^2 + dz^2);
615
616     % 6.4.3 Masa total de la viga
617     m_vig = rho * A_vig * Lv;
618
619     % 6.4.4 Masa concentrada en cada nodo (50%-50%)
620     m_nodo = m_vig / 2;
621

```

```

622 % 6.4.5 GDL de traslación de cada nodo
623 gdl_i = 6 * (ni - 1) + (1:3);
624 gdl_j = 6 * (nj - 1) + (1:3);
625
626 % 6.4.5 Agregar masa a diagonal
627 for d = 1:3
628     Mglobal(gdl_i(d), gdl_i(d)) = Mglobal(gdl_i(d), gdl_i(d)) + m_nodo;
629     Mglobal(gdl_j(d), gdl_j(d)) = Mglobal(gdl_j(d), gdl_j(d)) + m_nodo;
630 end
631 end
632
633 fprintf(' [OK] Masa de %d vigas agregada\n', size(E_vig, 1));
634
635 for e = 1:size(E_col, 1)
636     % Nodos extremos
637     ni = E_col(e, 2);
638     nj = E_col(e, 3);
639
640     % Longitud de columna
641     dx = N(nj).x - N(ni).x;
642     dy = N(nj).y - N(ni).y;
643     dz = N(nj).z - N(ni).z;
644     Lc = sqrt(dx^2 + dy^2 + dz^2);
645
646     % Masa total de la columna
647     m_col = rho * A_col * Lc;
648
649     % Masa concentrada en cada nodo (50%-50%)
650     m_nodo = m_col / 2;
651
652     % GDL de traslación de cada nodo
653     gdl_i = 6 * (ni - 1) + (1:3); % UX, UY, UZ del nodo i
654     gdl_j = 6 * (nj - 1) + (1:3); % UX, UY, UZ del nodo j
655
656     % Agregar masa a diagonal de Mglobal
657     for d = 1:3
658         Mglobal(gdl_i(d), gdl_i(d)) = Mglobal(gdl_i(d), gdl_i(d)) + m_nodo;
659         Mglobal(gdl_j(d), gdl_j(d)) = Mglobal(gdl_j(d), gdl_j(d)) + m_nodo;
660     end
661 end
662
663 fprintf(' [OK] Masa de %d columnas agregada\n', size(E_col, 1));
664
665 % 6.5 MASA ROTACIONAL (Sección Opcional)
666
667 fprintf(' Agregando inercia rotacional...\n');
668
669 % 6.5.1 Parámetros para cálculo de inercia (aproximación)
670 b_eq = b_col; % Base equivalente
671 r_giro = b_eq / sqrt(12); % Radio de giro para sección rectangular
672 m_col_media = rho * A_col * (h / 2); % Masa de media columna
673
674 % 6.5.2 Momento de inercia de masa respecto a eje Z
675 % Para sección cuadrada: I_zz = m * (b^2 + h^2) / 12
676 I_rz = m_col_media * (b_col^2 + h_col^2) / 12;

```

```

677
678 % 6.5.3 Contador de nodos con inercia agregada
679 n_inercias = 0;
680
681 for i = 1:nN
682     % GDL de interés
683     gdl_ux = 6 * (i - 1) + 1; % Traslación X
684     gdl_rz = 6 * (i - 1) + 6; % Rotación Z
685
686     % Solo agregar inercia si:
687     % 1. El nodo tiene masa traslacional (participó en elementos)
688     % 2. Aún no tiene inercia rotacional asignada
689     tiene_masa = Mglobal(gdl_ux, gdl_ux) > 1e-6;
690     sin_inercia = Mglobal(gdl_rz, gdl_rz) < 1e-10;
691
692     if tiene_masa && sin_inercia
693         Mglobal(gdl_rz, gdl_rz) = I_rz;
694         n_inercias = n_inercias + 1;
695     end
696 end
697
698 fprintf(' [OK] Inercia rotacional agregada a %d nodos\n', n_inercias);
699
700 % 6.6 MASA DE LOSAS
701 % Distribuye masa de losas según áreas tributarias de cada nodo
702
703 fprintf(' Agregando masa de losas...\n');
704
705 % 6.6.1 Parámetros de losas
706 espesor_losa = 0.25; % [m] Espesor de losa (vigueta-bovedilla)
707 rho_losa = rho; % [kg/m^3] Misma densidad que concreto
708 CM_adicional = 0; % [kg/m^2] Carga muerta adicional (acabados, instalaciones o
709     0)
710
711 % 6.6.2 Peso superficial de losa
712 peso_losa = espesor_losa * rho_losa; % [kg/m^2] Peso propio
713 masa_superficial_losa = peso_losa + CM_adicional; % [kg/m^2] Total
714
715 fprintf(' Losa: espesor=%.0f cm, masa=%.0f kg/m^2\n', ...
716     espesor_losa*100, masa_superficial_losa);
717
718 % 6.6.3 Contadores
719 masa_losas_agregada = 0;
720 n_nodos_con_losa = 0;
721
722 % 6.6.4 Recorrer todos los nodos EXCEPTO los de la base (z=0)
723 for iP = 1:(nProf + 1)
724     for iC = 1:(nCruj + 1)
725         for iNiv = 2:(nNiv + 1) % Desde nivel 1 hasta azotea (NO base)
726             % Índice del nodo
727             idx = nodeIndex(iNiv, iC, iP, nNiv, nCruj);
728
729             % Calcular área tributaria según posición en planta
730

```

```

731     % Detectar si está en bordes
732     en_borde_X_min = (iP == 1);
733     en_borde_X_max = (iP == nProf + 1);
734     en_borde_Y_min = (iC == 1);
735     en_borde_Y_max = (iC == nCruj + 1);
736
737     % Determinar área tributaria
738     if (en_borde_X_min || en_borde_X_max) && (en_borde_Y_min ||
739         en_borde_Y_max)
740         % NODO ESQUINA: 1/4 de celda
741         A_trib = (w / 2) * (L / 2);
742
743     elseif (en_borde_X_min || en_borde_X_max)
744         % NODO BORDE PARALELO A X: 1/2 celda en X, completa en Y
745         A_trib = (w / 2) * L;
746
747     elseif (en_borde_Y_min || en_borde_Y_max)
748         % NODO BORDE PARALELO A Y: completa en X, 1/2 celda en Y
749         A_trib = w * (L / 2);
750
751     else
752         % NODO INTERIOR: celda completa
753         A_trib = w * L;
754     end
755
756     % Masa de losa para este nodo
757     m_losa_nodo = A_trib * masa_superficial_losa;
758
759     % GDL de traslación del nodo
760     gdl_i = 6 * (idx - 1) + (1:3); % UX, UY, UZ
761
762     % Agregar masa de losa a las 3 direcciones traslacionales
763     for d = 1:3
764         Mglobal(gdl_i(d), gdl_i(d)) = Mglobal(gdl_i(d), gdl_i(d)) +
765             m_losa_nodo;
766     end
767
768     % Acumular estadísticas
769     masa_losas_agregada = masa_losas_agregada + m_losa_nodo;
770     n_nodos_con_losa = n_nodos_con_losa + 1;
771
772 end
773
774 end
775
776 end
777
778 fprintf(' [OK] Masa de losas agregada a %d nodos\n', n_nodos_con_losa);
779 fprintf(' [OK] Masa total de losas: %.0f kg (%.1f ton)\n', ...
780     masa_losas_agregada, masa_losas_agregada/1000);
781
782 % 6.7 Verificación: masa total de losas debería ser aproximadamente
783 % area_planta - n_niveles - masa_superficial
784 area_planta = (nProf * w) * (nCruj * L);
785 masa_teorica_losas = area_planta * nNiv * masa_superficial_losa;
786 error_losas = abs(masa_losas_agregada - masa_teorica_losas) / masa_teorica_losas
787     * 100;

```

```

783
784 fprintf(' [OK] Verificación: error = %.2f%% (esperado < 0.1%%)\n', error_losas)
    ;
785
786 if error_losas > 0.5
787     warning('Error en distribución de masa de losas > 0.5%%');
788 end
789
790 % 6.8 CALCULAR MASA TOTAL POR NODO (para visualización)
791
792 fprintf(' Calculando masa total por nodo...\n');
793
794 masa_nodal = zeros(nN, 1);
795
796 for i = 1:nN
797     % 6.8.1 GDL de traslación del nodo i
798     gdl_ux = 6 * (i - 1) + 1;
799     gdl_uy = 6 * (i - 1) + 2;
800     gdl_uz = 6 * (i - 1) + 3;
801
802     % 6.8.2 Suma de masas traslacionales (diagonal de M)
803     m_x = Mglobal(gdl_ux, gdl_ux);
804     m_y = Mglobal(gdl_uy, gdl_uy);
805     m_z = Mglobal(gdl_uz, gdl_uz);
806
807     % Masa total del nodo
808     % INCORRECTO (tu código actual):
809
810     % CORRECTO:
811     masa_nodal(i) = full(m_x); % Solo una dirección
812     % O equivalente:
813     %masa_nodal(i) = full(m_y);
814     %masa_nodal(i) = full(m_z);
815 end
816
817 % 6.9 Estadísticas (excluir nodos sin masa)
818 nodos_con_masa = masa_nodal > 0;
819 masa_min = min(masa_nodal(nodos_con_masa));
820 masa_max = max(masa_nodal);
821 masa_promedio = mean(masa_nodal(nodos_con_masa));
822 masa_total = sum(masa_nodal);
823
824 fprintf(' [OK] Masa nodal calculada\n');
825 fprintf(' Mínima: %.1f kg\n', masa_min);
826 fprintf(' Máxima: %.1f kg\n', masa_max);
827 fprintf(' Promedio: %.1f kg\n', masa_promedio);
828 fprintf(' TOTAL: %.0f kg (%.1f ton)\n\n', masa_total, masa_total/1000);
829
830
831 %% 7. Visualización de Distribución de Masas
832
833 fprintf('Generando gráfica de distribución de masas...\n');
834
835 % 7.1 Crear figura
836 figure('Name', 'Distribución de Masa Nodal', ...

```

```

837     'NumberTitle', 'off', ...
838     'Position', [100, 100, 800, 600]);
839
840 % 7.2 Calcular tamaños de marcadores (proporcional a masa)
841 masa_max = max(masa_nodal);
842 tamaño = 60 + 500 * (masa_nodal / masa_max); % Entre 60 y 560
843
844 % 7.3 Scatter 3D con color según masa
845 scatter3(xyz(:, 1), xyz(:, 2), xyz(:, 3), ...
846         tamaño, ...
847         masa_nodal, ...
848         'filled', ...
849         'MarkerEdgeColor', 'k', ...
850         'LineWidth', 0.5);
851
852 % 7.4 Configurar colorbar
853 c = colorbar;
854 c.Label.String = 'Masa [kg]';
855 c.Label.FontSize = 12;
856
857 % 7.5 Etiquetas
858 title('Distribución de Masa Nodal (Traslacional)', ...
859       'FontSize', 14, 'FontWeight', 'bold');
860 xlabel('X [m]', 'FontSize', 12);
861 ylabel('Y [m]', 'FontSize', 12);
862 zlabel('Z [m]', 'FontSize', 12);
863
864 % 7.6 Vista y aspecto
865 view(40, 30);
866 grid on;
867 axis equal;
868
869 fprintf(' [OK] Gráfica generada\n\n');
870
871 %% 8. Implementar diafragma rígido por piso (condicional)
872
873 if USE_DIAPHRAGM
874     fprintf('Implementando diafragmas rígidos...\n');
875
876 % 8.1 Para cada nivel (excepto base)
877 for iNiv = 2:(nNiv + 1) % Nivel 1 = primer piso
878
879     % 8.1.1 Encontrar todos los nodos en este nivel
880     nodos_piso = [];
881     coords_x = [];
882     coords_y = [];
883
884     for i = 1:nN
885         if abs(N(i).z - (iNiv-1)*h) < 1e-6 % Mismo nivel
886             nodos_piso = [nodos_piso, i];
887             coords_x = [coords_x, N(i).x];
888             coords_y = [coords_y, N(i).y];
889         end
890     end
891

```

```

892 % 8.1.2 Centro de masa del piso (nodo maestro)
893 xc = mean(coords_x);
894 yc = mean(coords_y);
895
896 % 8.1.3 Seleccionar nodo maestro (el más cercano al centro)
897 distancias = sqrt((coords_x - xc).^2 + (coords_y - yc).^2);
898 [~, idx_master] = min(distancias);
899 nodo_master = nodos_piso(idx_master);
900
901 % 8.1.4 Guardar información del diafragma
902 Diafragmas(iNiv-1).nivel = iNiv - 1;
903 Diafragmas(iNiv-1).nodo_master = nodo_master;
904 Diafragmas(iNiv-1).nodos_slave = setdiff(nodos_piso, nodo_master);
905 Diafragmas(iNiv-1).xc = xc;
906 Diafragmas(iNiv-1).yc = yc;
907
908 fprintf(' Nivel %d: Nodo master = %d, Slaves = %d\n', ...
909         iNiv-1, nodo_master, length(Diafragmas(iNiv-1).nodos_slave));
910 end
911
912 fprintf(' %d diafragmas definidos\n\n', length(Diafragmas));
913
914 else
915 % Sin diafragma: notificar y continuar sin condensación
916 fprintf('Diafragma rígido: DESACTIVADO (USE_DIAPHRAGM = false)\n');
917 fprintf(' Los shells de membrana contribuyen activamente a K_red.\n');
918 fprintf(' Los nodos de cada piso son cinemáticamente independientes.\n\n');
919 % Diafragmas no se define -> mrigidez.m ejecutará rama else de la Sec. 4
920 end

```

B.2. mrigidez.m

```

1 %% Ensamblaje de Matriz de Rigidez
2 % SCRIPT: ENSAMBLAJE DE MATRIZ DE RIGIDEZ GLOBAL CON DIAFRAGMA RÍGIDO
3 % Genera matriz K para marco 3D con 6 GDL por nodo
4 % con diafragma rígido por piso (Master-Slave)
5
6 %% 1. Inicialización
7 fprintf(' ENSAMBLAJE DE MATRIZ DE RIGIDEZ GLOBAL\n');
8 fprintf('-----\n\n');
9
10 nBarras = length(Barras);
11
12 fprintf('Información del modelo:\n');
13 fprintf(' Nodos: %d\n', nN);
14 fprintf(' Barras: %d\n', nBarras);
15 fprintf(' GDL: %d\n\n', ngdl);
16
17 % 1.1 Inicializar matriz de rigidez global (sparse para eficiencia)
18 Kglobal = sparse(ngdl, ngdl);
19
20 fprintf('Inicializando matrices...\n');

```

```

21 fprintf(' Matriz K: %d x %d (sparse)\n\n', ngdl, ngdl);
22
23
24 %% 2. Ensamblaje por elementos
25 fprintf('Ensamblando matriz de rigidez...\n');
26
27 % 2.1 Contadores para diagnóstico
28 n_columnas_ensambladas = 0;
29 n_vigas_ensambladas = 0;
30
31 for e = 1:nBarras
32
33 % 2.2 Información del elemento
34 % 2.2.1 Nodos extremos
35 ni = Barras(e).nodo1;
36 nj = Barras(e).nodo2;
37
38 % 2.2.2 Coordenadas
39 xi = N(ni).x; yi = N(ni).y; zi = N(ni).z;
40 xj = N(nj).x; yj = N(nj).y; zj = N(nj).z;
41
42 % 2.2.3 Longitud (usar la ya calculada)
43 L_elem = Barras(e).L;
44
45 % 2.3 Determinar propiedades según tipo
46 if strcmp(Barras(e).tipo, 'columna')
47 % 2.3.1 COLUMNA
48 Ae = A_col;
49 Iy = Iy_col;
50 Iz = Iz_col;
51 Je = J_col;
52 n_columnas_ensambladas = n_columnas_ensambladas + 1;
53 else
54 % 2.3.2 VIGA (X o Y)
55 Ae = A_vig;
56 Iy = Iy_vig;
57 Iz = Iz_vig;
58 Je = J_vig;
59 n_vigas_ensambladas = n_vigas_ensambladas + 1;
60 end
61
62 % 2.4 Matriz de rigidez local (12x12)
63 kLocal = localStiffness3D(E, G, Je, Ae, Iy, Iz, L_elem);
64
65 % 2.5 Matriz de transformación (12x12)
66 T = transformMatrix3D(xi, yi, zi, xj, yj, zj);
67
68 % 2.6 Transformar a coordenadas globales
69
70 kGlobal_elem = T' * kLocal * T; % ORDEN CORRECTO: T' x K x T
71
72 % 2.7 GDL del elemento en matriz global
73 dofs_i = 6 * (ni - 1) + (1:6); % GDL del nodo i
74 dofs_j = 6 * (nj - 1) + (1:6); % GDL del nodo j
75 dofs_elem = [dofs_i, dofs_j]; % 12 GDL del elemento

```

```

76
77 % 2.8 Ensamblar en matriz global
78 Kglobal(dofs_elem, dofs_elem) = Kglobal(dofs_elem, dofs_elem) + kGlobal_elem;
79 end
80
81 fprintf(' [OK] %d columnas ensambladas\n', n_columnas_ensambladas);
82 fprintf(' [OK] %d vigas ensambladas\n', n_vigas_ensambladas);
83 fprintf(' [OK] Total: %d elementos\n\n', nBarras);
84
85 %% 2b. Ensamblaje de elementos SHELL de membrana (losas)
86 % Elemento Q4 bilineal de 4 nodos, estado plano de esfuerzos (plane stress)
87 % Replica los 192 shell elements de SAP2000 (sección LOSA 25CM-250)
88 % Solo DOF de membrana: Ux y Uy de los 4 nodos esquina de cada panel
89 % El diafragma rígido se aplica ENCIMA de esta rigidez (Sección 4)
90 % Referencia: Cook, Malkus, Plesha - Concepts and Applications of FEA, Cap. 6
91
92 fprintf('Ensamblando elementos shell de membrana (losas)...\n');
93
94 % 2b.1 Parámetros del elemento shell
95 E_shell = E; % [Pa] Módulo de elasticidad (mismo que el marco)
96 nu_shell = 0.3; % Coeficiente de Poisson del concreto
97 t_shell = 0.25; % [m] Espesor de losa (LOSA 25CM en SAP2000)
98
99 % 2b.2 Semi-dimensiones del panel rectangular
100 % Todos los paneles son iguales para esta geometría regular
101 a_shell = w / 2; % [m] Semi-dimensión en X (w = separación entre ejes en X)
102 b_shell = L / 2; % [m] Semi-dimensión en Y (L = separación entre ejes en Y)
103
104 % 2b.3 Calcular K_membrana una sola vez (todos los paneles son idénticos)
105 K_mem = membraneStiffnessQ4(E_shell, nu_shell, t_shell, a_shell, b_shell);
106
107 % Factor de reducción de rigidez de membrana (calibrado para T1 = T_SAP2000)
108 % Físicamente equivale a los modificadores de rigidez de SAP2000 (f11=f22=f12=
109 % Justificación: ACI 318-19 Tabla 6.6.3.1.1(a) permite 0.25*Ig para losas bajo
110 % Aquí se usa alpha=0.302 calibrado para que T1_MATLAB = T1_SAP2000 = 1.788 s.
111 % Ref: Aviles et al. (2016), Cap. 6 - modelo de viga de cortante con rigidez
112 % lateral
113 % calibrada; NTC-Sismo (2020), Sec. 5.2 - reducción de rigidez por fisurado.
114 alpha_mem = 0.302; % Factor de reducción de membrana [-]
115 fprintf(' Factor reducción membrana: alpha = %.3f\n', alpha_mem);
116
117 % 2b.4 Contadores y acumuladores
118 n_shells = 0;
119
120 % 2b.5 Recorrer cada panel: nivel x crujía_X x crujía_Y
121 for iNiv = 2:(nNiv + 1) % Niveles 1 a nNiv (excluye base z=0)
122     for iP = 1:nProf % Posición en X (crujías)
123         for iC = 1:nCruj % Posición en Y (crujías)
124
125             % 2b.5.1 Índices de los 4 nodos esquina del panel
126             % Numeración en sentido antihorario vista desde arriba (+Z)
127             % Coincide con convención SAP2000 para area elements
128             n1 = nodeIndex(iNiv, iC, iP, nNiv, nCruj); % esquina (iP, iC )

```

```

128     n2 = nodeIndex(iNiv, iC, iP+1, nNiv, nCruj); % esquina (iP+1, iC )
129     n3 = nodeIndex(iNiv, iC+1, iP+1, nNiv, nCruj); % esquina (iP+1, iC+1)
130     n4 = nodeIndex(iNiv, iC+1, iP, nNiv, nCruj); % esquina (iP, iC+1)
131     nodos_shell = [n1, n2, n3, n4];
132
133     % 2b.5.2 DOF globales de membrana del elemento (8 DOF)
134     % Orden: [Ux_n1, Uy_n1, Ux_n2, Uy_n2, Ux_n3, Uy_n3, Ux_n4, Uy_n4]
135     % Solo Ux (DOF 1) y Uy (DOF 2) de cada nodo participan en membrana
136     dofs_mem = zeros(1, 8);
137     for k = 1:4
138         nd = nodos_shell(k);
139         dofs_mem(2*k-1) = 6*(nd-1) + 1; % Ux del nodo k
140         dofs_mem(2*k) = 6*(nd-1) + 2; % Uy del nodo k
141     end
142
143     % 2b.5.3 Ensamblar K_membrana en la matriz global
144     Kglobal(dofs_mem, dofs_mem) = Kglobal(dofs_mem, dofs_mem) + alpha_mem *
        K_mem;
145
146     n_shells = n_shells + 1;
147 end
148 end
149 end
150
151 fprintf(' Shells ensamblados: %d (esperado: %d)\n', n_shells, nProf*nCruj*nNiv);
152 fprintf(' E_shell = %.3f GPa\n', E_shell/1e9);
153 fprintf(' nu = %.2f\n', nu_shell);
154 fprintf(' Espesor = %.2f m\n', t_shell);
155 fprintf(' Panel: %.1f m x %.1f m\n\n', w, L);
156
157
158 %% 2c. [DESACTIVADO] Ensamblaje shell de flexión Mindlin Q4
159 % No se incluye: la membrana reducida con alpha=0.302 es suficiente.
160 % % Componente de flexión fuera del plano (bending) del thin shell.
161 % % Se combina con la membrana Q4 (sección 2b) para reproducir el
162 % % comportamiento de thin shell de SAP2000 (membrane + bending).
163 % %
164 % % Teoría: Mindlin (First-Order Shear Deformation Theory, FSDT)
165 % % Supuesto: normales a la superficie media permanecen rectas pero NO
166 % % necesariamente normales a la superficie deformada (incluye cortante
167 % % transversal, a diferencia de Kirchhoff).
168 % %
169 % % DOF por nodo: w (Uz, desplazamiento vertical), theta_x (Rx), theta_y (Ry)
170 % % DOF del elemento: [w1,Rx1,Ry1, w2,Rx2,Ry2, w3,Rx3,Ry3, w4,Rx4,Ry4] (12 DOF)
171 % %
172 % % En el sistema global 6-DOF por nodo: Uz=DOF3, Rx=DOF4, Ry=DOF5
173 % %
174 % % Integración selectiva para evitar shear locking (Ferreira, 2009):
175 % % - Rigidez de flexión (K_bend): cuadratura 2x2 Gauss
176 % % - Rigidez de cortante transversal (K_shear): cuadratura reducida 1x1
177 % %
178 % % Referencia: Ferreira, A.J.M. (2009). MATLAB Codes for Finite Element
179 % % Analysis. Springer. Cap. 12, Ec. 12.23-12.24.
180 % % Factor de corrección de cortante kappa=5/6: Mindlin (1951).
181

```

```

182 % fprintf('Ensamblando elementos shell de flexión Mindlin Q4 (losas)...\n');
183
184 % % 2c.1 Parámetros constitutivos
185 % % Mismos E_shell, nu_shell, t_shell definidos en sección 2b
186 % kappa = 5/6; % Factor de corrección de cortante
187 % G_shell = E_shell / (2*(1 + nu_shell)); % Módulo de cortante [Pa]
188
189 % % 2c.2 Matrices constitutivas de placa (Ferreira, 2009, Ec. 12.13, 12.15)
190 % % Rigidez de flexión: D_f = E*h^3/(12*(1-nu^2)) * [constitutiva]
191 % D_f = (E_shell * t_shell^3 / (12*(1 - nu_shell^2))) * ...
192 % [1, nu_shell, 0;
193 % nu_shell, 1, 0;
194 % 0, 0, (1-nu_shell)/2];
195
196 % % Rigidez de cortante transverso: D_c = kappa*h*G * I_2x2
197 % D_c = kappa * t_shell * G_shell * eye(2);
198
199 % % 2c.3 Calcular K_bend una sola vez (todos los paneles son idénticos)
200 % K_bend = mindlinBendingQ4(D_f, D_c, a_shell, b_shell);
201
202 % % 2c.4 Ensamblaje en Kglobal
203 % n_bend = 0;
204
205 % for iNiv = 2:(nNiv + 1)
206 % for iP = 1:nProf
207 % for iC = 1:nCruj
208
209 % % Mismos 4 nodos que la membrana (misma numeración)
210 % n1 = nodeIndex(iNiv, iC, iP, nNiv, nCruj);
211 % n2 = nodeIndex(iNiv, iC, iP+1, nNiv, nCruj);
212 % n3 = nodeIndex(iNiv, iC+1, iP+1, nNiv, nCruj);
213 % n4 = nodeIndex(iNiv, iC+1, iP, nNiv, nCruj);
214 % nodos_b = [n1, n2, n3, n4];
215
216 % % DOF globales de flexión del elemento (12 DOF)
217 % % Orden local: [w1, Rx1, Ry1, w2, Rx2, Ry2, w3, Rx3, Ry3, w4, Rx4, Ry4]
218 % % En sistema global 6-DOF/nodo: Uz=3, Rx=4, Ry=5
219 % dofs_b = zeros(1, 12);
220 % for k = 1:4
221 % nd = nodos_b(k);
222 % base = 6*(nd-1);
223 % dofs_b(3*k-2) = base + 3; % Uz (w)
224 % dofs_b(3*k-1) = base + 4; % Rx (theta_x)
225 % dofs_b(3*k) = base + 5; % Ry (theta_y)
226 % end
227
228 % % Ensamblar K_bend en la matriz global
229 % Kglobal(dofs_b, dofs_b) = Kglobal(dofs_b, dofs_b) + K_bend;
230
231 % n_bend = n_bend + 1;
232 % end
233 % end
234 % end
235
236 % fprintf(' Flexión Mindlin ensamblada: %d paneles\n', n_bend);

```

```

237 % fprintf(' D_f(1,1) = %.3e N*m\n', D_f(1,1));
238 % fprintf(' D_c(1,1) = %.3e N/m\n', D_c(1,1));
239 % fprintf(' kappa = %.4f\n\n', kappa);
240
241
242 %% 3. Verificación de Matriz K (antes del diafragma)
243
244 fprintf('Verificando propiedades de la matriz K...\n');
245
246 % 3.1 Simetría
247 error_simetria = norm(Kglobal - Kglobal', 'fro') / norm(Kglobal, 'fro');
248 fprintf(' Simetría: error = %.2e (debe ser < 1e-10)\n', error_simetria);
249
250 if error_simetria > 1e-8
251     warning('Matriz K no es simétrica. Revisar ensamblaje.');
```

end

```

252
253
254 % 3.2 Esparsidad
255 n_nonzero = nnz(Kglobal);
256 n_total = ngdl^2;
257 esparsidad = (1 - n_nonzero / n_total) * 100;
258 fprintf(' Elementos no cero: %d de %d (%.1f%% sparse)\n', ...
259     n_nonzero, n_total, esparsidad);
260
261 % 3.3 Diagonal positiva
262 diagonal_K = diag(Kglobal);
263 n_positivos = full(sum(diagonal_K > 0));
264 fprintf(' Diagonal: %d/%d elementos positivos\n\n', n_positivos, ngdl);
265
266 %% 4. Aplicar diafragma rígido
267 if exist('Diafragmas', 'var') && ~isempty(Diafragmas)
268     fprintf(' APLICANDO DIAFRAGMA RÍGIDO (Master-Slave)\n');
269     fprintf('-----\n\n');
```

% 4.1 Identificar GDL dependientes

```

272     fprintf('Identificando GDL dependientes...\n');
273
274     gdl_dependientes = []; % GDL que dependen del maestro
275
276     for d = 1:length(Diafragmas)
277         nodos_s = Diafragmas(d).nodos_slave;
278
279         for ns = nodos_s
280             % Los GDL Ux, Uy, Rz del esclavo dependen del maestro
281             gdl_ux = 6*(ns-1) + 1;
282             gdl_uy = 6*(ns-1) + 2;
283             gdl_rz = 6*(ns-1) + 6;
284
285             gdl_dependientes = [gdl_dependientes, gdl_ux, gdl_uy, gdl_rz];
286         end
287     end
288
289     gdl_dependientes = unique(gdl_dependientes);
290
291     fprintf(' GDL dependientes: %d\n', length(gdl_dependientes));
```

```

292 fprintf(' GDL originales: %d\n', ngdl);
293
294
295 % 4.2 Construir matriz de transformación
296
297 fprintf(' Construyendo matriz de transformación...\n');
298
299 % 4.3 GDL independientes (los que quedan)
300 todos_gdl = 1:ngdl;
301 gdl_independientes = setdiff(todos_gdl, gdl_dependientes);
302
303 % 4.4 Matriz de transformación T: u_total = T x u_reducido
304 n_indep = length(gdl_independientes);
305 T_diafragma = sparse(ngdl, n_indep);
306
307 % 4.5 Inicializar identidad para GDL independientes
308 for i = 1:n_indep
309     T_diafragma(gdl_independientes(i), i) = 1;
310 end
311
312 % 4.6 Agregar relaciones de dependencia (restricciones cinemáticas)
313 for d = 1:length(Diafragmas)
314     nodo_m = Diafragmas(d).nodo_master;
315     nodos_s = Diafragmas(d).nodos_slave;
316     xc = Diafragmas(d).xc;
317     yc = Diafragmas(d).yc;
318
319     % 4.6.1 GDL del maestro
320     gdl_m_ux = 6*(nodo_m-1) + 1;
321     gdl_m_uy = 6*(nodo_m-1) + 2;
322     gdl_m_rz = 6*(nodo_m-1) + 6;
323
324     % 4.6.2 Encontrar columnas en T
325     col_m_ux = find(gdl_independientes == gdl_m_ux);
326     col_m_uy = find(gdl_independientes == gdl_m_uy);
327     col_m_rz = find(gdl_independientes == gdl_m_rz);
328
329     for ns = nodos_s
330         % Distancia al centro geométrico del piso
331         dx = N(ns).x - xc;
332         dy = N(ns).y - yc;
333
334         % GDL del esclavo
335         gdl_s_ux = 6*(ns-1) + 1;
336         gdl_s_uy = 6*(ns-1) + 2;
337         gdl_s_rz = 6*(ns-1) + 6;
338
339         % Relaciones cinemáticas del diafragma rígido:
340         % Ux_slave = Ux_master - Rz_master x dy
341         T_diafragma(gdl_s_ux, col_m_ux) = 1;
342         T_diafragma(gdl_s_ux, col_m_rz) = -dy;
343
344         % Uy_slave = Uy_master + Rz_master x dx
345         T_diafragma(gdl_s_uy, col_m_uy) = 1;
346         T_diafragma(gdl_s_uy, col_m_rz) = dx;

```

```

347         % Rz_slave = Rz_master
348         T_diafragma(gdl_s_rz, col_m_rz) = 1;
349     end
350 end
351 end
352
353 fprintf(' [OK] Matriz T: %d [OK]? %d GDL\n', ngdl, n_indep);
354
355
356 % 4.7 Transformar matrices K y M
357 fprintf(' Transformando matrices K y M...\n');
358
359 % K_diafragma = T' x K x T
360 % M_diafragma = T' x M x T
361 Kglobal_original = Kglobal; % Guardar copia
362 Mglobal_original = Mglobal;
363
364 Kglobal = T_diafragma' * Kglobal * T_diafragma;
365 Mglobal = T_diafragma' * Mglobal * T_diafragma;
366
367 % 4.8 Actualizar tamaño de ngdl
368 ngdl_original = ngdl;
369 ngdl = n_indep;
370
371 fprintf(' [OK] K: %d x %d [OK]? %d x %d\n', ngdl_original, ngdl_original,
372         ngdl, ngdl);
373 fprintf(' [OK] M: %d x %d [OK]? %d x %d\n', ngdl_original, ngdl_original,
374         ngdl, ngdl);
375 fprintf(' [OK] Reducción: %.1f%% de GDL eliminados\n\n', ...
376         (1 - ngdl/ngdl_original) * 100);
377
378 % 4.9 Guardar mapeo para post-procesamiento
379 gdl_mapping.independientes = gdl_independientes;
380 gdl_mapping.dependientes = gdl_dependientes;
381 gdl_mapping.T = T_diafragma;
382 gdl_mapping.ngdl_original = ngdl_original;
383 gdl_mapping.ngdl_reducido = ngdl;
384
385 fprintf('Diafragmas aplicados exitosamente:\n');
386 fprintf(' %d pisos con diafragma rígido\n', length(Diafragmas));
387 fprintf(' GDL eliminados: %d\n', length(gdl_dependientes));
388 fprintf(' GDL restantes: %d\n\n', ngdl);
389
390 else
391     fprintf('Variable Diafragmas no encontrada');
392     fprintf('Ejecutando SIN diafragma rígido\n\n');
393 end
394
395 %% 5. Aplicar condiciones de frontera
396
397 fprintf('Aplicando condiciones de frontera...\n');
398
399 % 5.1 Identificar GDL restringidos (nodos en z=0, empotrados)
400 gdl_restringidos = [];

```

```

400
401 if exist('gdl_independientes', 'var')
402     % 5.1.1 CON DIAFRAGMA: Solo restringir GDL que siguen siendo independientes
403     for i = 1:nN
404         if N(i).z == 0 % Nodo en la base
405             gdl_base = 6 * (i - 1) + (1:6); % Todos los 6 GDL del nodo
406             % Solo agregar los que siguen siendo independientes
407             gdl_restringidos = [gdl_restringidos, ...
408                 intersect(gdl_base, gdl_independientes)];
409         end
410     end
411 else
412     % 5.1.2 SIN DIAFRAGMA: Restringir todos los GDL de la base
413     for i = 1:nN
414         if N(i).z == 0 % Nodo en la base
415             gdl_base = 6 * (i - 1) + (1:6);
416             gdl_restringidos = [gdl_restringidos, gdl_base];
417         end
418     end
419 end
420
421 gdl_restringidos = unique(gdl_restringidos);
422
423 % 5.2 GDL libres
424 % NOTA: gdl_restringidos contiene índices del espacio ORIGINAL (1..1950).
425 % ngdl es el tamaño del espacio POST-diafragma (1086). Para evitar mezclar
426 % los dos espacios de indexación, se convierten los índices originales a
427 % posiciones en gdl_independientes antes de aplicar setdiff.
428 if exist('gdl_independientes', 'var')
429     pos_restringidos = [];
430     for g = gdl_restringidos
431         p = find(gdl_independientes == g, 1);
432         if ~isempty(p)
433             pos_restringidos = [pos_restringidos, p];
434         end
435     end
436     gdl_libres = setdiff(1:ngdl, pos_restringidos);
437 else
438     gdl_libres = setdiff(1:ngdl, gdl_restringidos);
439 end
440
441 fprintf(' GDL restringidos (empotramientos): %d\n', length(gdl_restringidos));
442 fprintf(' GDL libres: %d\n\n', length(gdl_libres));
443
444 % 5.3 Matrices reducidas (solo GDL libres)
445 K_red = Kglobal(gdl_libres, gdl_libres);
446 M_red = Mglobal(gdl_libres, gdl_libres);
447
448 fprintf('Matrices reducidas generadas:\n');
449 fprintf(' Tamaño: %d x %d\n', length(gdl_libres), length(gdl_libres));
450 fprintf(' K_red elementos no cero: %d\n', nnz(K_red));
451 fprintf(' M_red elementos no cero: %d\n\n', nnz(M_red));
452
453
454 %% 6. Verificación de definición positiva

```

```

455 fprintf('Verificando definición positiva de K_red...\n');
456
457 % 6.1 Calcular autovalores mínimo y máximo
458 lambda_min = eigs(K_red, 1, 'smallestabs');
459 lambda_max = eigs(K_red, 1, 'largestabs');
460
461 fprintf(' Autovalor mínimo: %.6e\n', lambda_min);
462 fprintf(' Autovalor máximo: %.6e\n', lambda_max);
463 fprintf(' Número de condición: %.2e\n\n', lambda_max / lambda_min);
464
465 if lambda_min < 0
466     warning('K_red tiene autovalores negativos. La estructura puede tener modos
         de cuerpo rígido.');
```

```

467 elseif lambda_min < 1e-6
468     warning('K_red está mal condicionada. Puede haber modos de cuerpo rígido no
         eliminados.');
```

```

469 else
470     fprintf(' [OK] K_red es definida positiva\n\n');
```

```

471 end
472
473
474 %% 7. Inicializar vector de fuerzas
475 % 7.1 Vector de fuerzas externas (para análisis estático/dinámico posterior)
476 F_ext = sparse(ngdl, 1);
477
478 fprintf('Vector de fuerzas externas inicializado (ceros)\n');
479 fprintf(' Tamaño: %d x 1\n\n', ngdl);
480
481
482 %% 8. Visualización de Matrices
483 fprintf('Generando gráficas de estructura de matrices...\n');
```

```

484
485 % 8.1 Matriz K completa (después de diafragma)
486 figure('Name', 'Matriz de Rigidez con Diafragma', ...
487     'NumberTitle', 'off', ...
488     'Position', [50, 50, 800, 700]);
489
490 spy(Kglobal);
491 title('Matriz de Rigidez Global K (con diafragma rígido)', ...
492     'FontSize', 14, 'FontWeight', 'bold');
```

```

493 xlabel('GDL j', 'FontSize', 12);
494 ylabel('GDL i', 'FontSize', 12);
495 grid on;
496
497 % 8.1.1 Añadir información
498 if exist('gdl_mapping', 'var')
499     text_info = sprintf('Original: %d x %d\nCon diafragma: %d x %d\nReducción:
         %.1f%%', ...
500         ngdl_original, ngdl_original, ...
501         ngdl, ngdl, ...
502         (1 - ngdl/ngdl_original)*100);
503 else
504     text_info = sprintf('Tamaño: %d x %d\nSin diafragma', ngdl, ngdl);
505 end
506
```

```

507 annotation('textbox', [0.15, 0.80, 0.3, 0.1], ...
508     'String', text_info, ...
509     'FitBoxToText', 'on', ...
510     'FontSize', 10, ...
511     'BackgroundColor', 'white', ...
512     'EdgeColor', 'black');
513
514 % 8.2 Matriz K reducida (sin empotramientos)
515 figure('Name', 'Matriz de Rigidez Reducida', ...
516     'NumberTitle', 'off', ...
517     'Position', [100, 100, 800, 700]);
518
519 spy(K_red);
520 title('Matriz de Rigidez Reducida K_{red} (sin empotramientos)', ...
521     'FontSize', 14, 'FontWeight', 'bold');
522 xlabel('GDL j', 'FontSize', 12);
523 ylabel('GDL i', 'FontSize', 12);
524 grid on;
525
526 n_gdl_red = length(gdl_libres);
527 text_info_red = sprintf('Tamaño: %d x %d\nGDL libres: %d', ...
528     n_gdl_red, n_gdl_red, n_gdl_red);
529 annotation('textbox', [0.15, 0.80, 0.3, 0.1], ...
530     'String', text_info_red, ...
531     'FitBoxToText', 'on', ...
532     'FontSize', 10, ...
533     'BackgroundColor', 'white', ...
534     'EdgeColor', 'black');
535
536 fprintf(' [OK] Gráficas generadas\n\n');
537
538
539 %% 9. Resumen Final
540 fprintf(' RESUMEN DE MATRICES GENERADAS\n');
541 fprintf('-----\n');
542
543 if exist('gdl_mapping', 'var')
544     fprintf('Diafragma rígido:\n');
545     fprintf(' Aplicado: SI\n');
546     fprintf(' Pisos: %d\n', length(Diafragmas));
547     fprintf(' GDL eliminados: %d\n\n', length(gdl_dependientes));
548 end
549
550 fprintf('Matriz de Rigidez:\n');
551 fprintf(' Kglobal: %d x %d (con diafragma)\n', ngdl, ngdl);
552 fprintf(' K_red: %d x %d (sin empotramientos)\n', n_gdl_red, n_gdl_red);
553 fprintf('\nMatriz de Masa:\n');
554 fprintf(' Mglobal: %d x %d (con diafragma)\n', ngdl, ngdl);
555 fprintf(' M_red: %d x %d (sin empotramientos)\n', n_gdl_red, n_gdl_red);
556 fprintf('\nVector de Fuerzas:\n');
557 fprintf(' F_ext: %d x 1 (inicializado en ceros)\n', ngdl);
558 fprintf('\nGDL:\n');
559 fprintf(' Total (con diafragma): %d\n', ngdl);
560 fprintf(' Restringidos: %d\n', length(gdl_restringidos));
561 fprintf(' Libres: %d\n', length(gdl_libres));

```

```

562 fprintf('-----\n\n');
563
564 fprintf('[OK] Ensamblaje completado exitosamente\n\n');
565
566 %% Funciones locales
567
568 % FUNCIÓN: localStiffness3D
569 % Matriz de rigidez local 12x12 para elemento viga-columna 3D
570
571 function k = localStiffness3D(E, G, J, A, Iy, Iz, L)
572     % Inicializar matriz 12x12
573     k = zeros(12, 12);
574
575     % Coeficientes
576     EA_L = E * A / L;
577     GJ_L = G * J / L;
578
579     EIy_L3 = E * Iy / L^3;
580     EIZ_L3 = E * Iz / L^3;
581
582     EIy_L2 = E * Iy / L^2;
583     EIZ_L2 = E * Iz / L^2;
584
585     EIy_L = E * Iy / L;
586     EIZ_L = E * Iz / L;
587
588     % RIGIDEZ AXIAL (GDL 1 y 7: u)
589     k(1, 1) = EA_L;
590     k(1, 7) = -EA_L;
591     k(7, 1) = -EA_L;
592     k(7, 7) = EA_L;
593
594     % RIGIDEZ TORSIONAL (GDL 4 y 10: thetax)
595     k(4, 4) = GJ_L;
596     k(4, 10) = -GJ_L;
597     k(10, 4) = -GJ_L;
598     k(10, 10) = GJ_L;
599
600     % FLEXIÓN EN PLANO XZ (GDL 2, 6, 8, 12: v, thetaz)
601     k(2, 2) = 12 * EIz_L3;
602     k(2, 6) = 6 * EIz_L2;
603     k(2, 8) = -12 * EIz_L3;
604     k(2, 12) = 6 * EIz_L2;
605
606     k(6, 2) = 6 * EIz_L2;
607     k(6, 6) = 4 * EIz_L;
608     k(6, 8) = -6 * EIz_L2;
609     k(6, 12) = 2 * EIz_L;
610
611     k(8, 2) = -12 * EIz_L3;
612     k(8, 6) = -6 * EIz_L2;
613     k(8, 8) = 12 * EIz_L3;
614     k(8, 12) = -6 * EIz_L2;
615
616     k(12, 2) = 6 * EIz_L2;

```

```

617 k(12, 6) = 2 * EIz_L;
618 k(12, 8) = -6 * EIz_L2;
619 k(12, 12) = 4 * EIz_L;
620
621 % FLEXIÓN EN PLANO XY (GDL 3, 5, 9, 11: w, thetay)
622 k(3, 3) = 12 * EIy_L3;
623 k(3, 5) = -6 * EIy_L2;
624 k(3, 9) = -12 * EIy_L3;
625 k(3, 11) = -6 * EIy_L2;
626
627 k(5, 3) = -6 * EIy_L2;
628 k(5, 5) = 4 * EIy_L;
629 k(5, 9) = 6 * EIy_L2;
630 k(5, 11) = 2 * EIy_L;
631
632 k(9, 3) = -12 * EIy_L3;
633 k(9, 5) = 6 * EIy_L2;
634 k(9, 9) = 12 * EIy_L3;
635 k(9, 11) = 6 * EIy_L2;
636
637 k(11, 3) = -6 * EIy_L2;
638 k(11, 5) = 2 * EIy_L;
639 k(11, 9) = 6 * EIy_L2;
640 k(11, 11) = 4 * EIy_L;
641 end
642
643
644 % FUNCIÓN: transformMatrix3D
645 % Matriz de transformación de coordenadas locales a globales
646
647 function T = transformMatrix3D(xi, yi, zi, xj, yj, zj)
648 % Vector del elemento (global)
649 dx = xj - xi;
650 dy = yj - yi;
651 dz = zj - zi;
652 L = sqrt(dx^2 + dy^2 + dz^2);
653
654 % Cosenos directores del eje local x (eje del elemento)
655 cx = dx / L;
656 cy = dy / L;
657 cz = dz / L;
658
659 % Determinar ejes locales y, z
660 tol = 1e-6;
661
662 if abs(cx) < tol && abs(cy) < tol
663 % ELEMENTO VERTICAL (columna)
664 vy = [1; 0; 0];
665 else
666 % ELEMENTO HORIZONTAL O INCLINADO
667 vz_ref = [0; 0; 1];
668 vx = [cx; cy; cz];
669 vy = cross(vz_ref, vx);
670 vy = vy / norm(vy);
671 end

```

```

672
673 % Eje local z = x x y
674 vx = [cx; cy; cz];
675 vz = cross(vx, vy);
676 vz = vz / norm(vz);
677
678 % Matriz de rotación 3x3
679 R = [vx, vy, vz]';
680
681 % Matriz de transformación 12x12
682 T = zeros(12, 12);
683 T(1:3, 1:3) = R;
684 T(4:6, 4:6) = R;
685 T(7:9, 7:9) = R;
686 T(10:12, 10:12) = R;
687 end
688
689
690 %% FUNCIÓN: nodeIndex
691 % Calcula índice lineal de nodo según posición (iN, iC, iP) en malla 3D
692 % Idéntica a la definida en marco3d_editable.m y analisis_dinamico.m
693 % Se incluye aquí para que mrigidez.m sea autocontenido
694
695 function idx = nodeIndex(iN, iC, iP, nNiv, nCruj)
696     offset_profundidad = (iP - 1) * (nCruj + 1) * (nNiv + 1);
697     offset_crujia = (iC - 1) * (nNiv + 1);
698     offset_nivel = iN;
699     idx = offset_profundidad + offset_crujia + offset_nivel;
700 end
701
702 %% FUNCIÓN: membraneStiffnessQ4
703 % Matriz de rigidez de membrana para elemento rectangular Q4 de 4 nodos
704 % Estado plano de esfuerzos (plane stress)
705 %
706 % INPUTS:
707 % E [Pa] Módulo de elasticidad
708 % nu [-] Coeficiente de Poisson
709 % t [m] Espesor del elemento
710 % a [m] Semi-dimensión en dirección local x (ancho total = 2a)
711 % b [m] Semi-dimensión en dirección local y (largo total = 2b)
712 %
713 % OUTPUT:
714 % K [8x8] Matriz de rigidez de membrana en coordenadas locales
715 % DOF: [Ux1,Uy1, Ux2,Uy2, Ux3,Uy3, Ux4,Uy4]
716 % Nodos en sentido antihorario: (-, -), (+, -), (+, +), (-, +)
717 %
718 % Para losa horizontal: local x = global X, local y = global Y
719 % No requiere transformación de coordenadas.
720 %
721 % Referencia: Cook, Malkus, Plesha & Witt (2002), Cap. 6, Ec. 6.5-4
722
723 function K = membraneStiffnessQ4(E, nu, t, a, b)
724
725     % Matriz constitutiva de plane stress
726     c = E / (1 - nu^2);

```

```

727 D = c * [1, nu, 0;
728         nu, 1, 0;
729         0, 0, (1-nu)/2];
730
731 % Inicializar K
732 K = zeros(8, 8);
733
734 % Integración numérica: cuadratura de Gauss 2x2 (exacta para Q4 bilineal)
735 gp = 1.0 / sqrt(3);
736 xi_pts = [-gp, gp, gp, -gp];
737 eta_pts = [-gp, -gp, gp, gp];
738 % Todos los pesos = 1 para cuadratura 2x2
739
740 for q = 1:4
741     xi = xi_pts(q);
742     eta = eta_pts(q);
743
744     % Derivadas de funciones de forma respecto a coordenadas naturales
745     % N1=(1-xi)(1-eta)/4, N2=(1+xi)(1-eta)/4,
746     % N3=(1+xi)(1+eta)/4, N4=(1-xi)(1+eta)/4
747     dN_dxi = [-(1-eta), (1-eta), (1+eta), -(1+eta)] / 4;
748     dN_deta = [-(1-xi), -(1+xi), (1+xi), (1-xi)] / 4;
749
750     % Jacobiano para elemento rectangular: J = diag([a, b])
751     % det(J) = a * b
752     detJ = a * b;
753
754     % Derivadas de funciones de forma respecto a coordenadas físicas
755     dN_dx = dN_dxi / a; % = Jinv(1,1) * dN_dxi
756     dN_dy = dN_deta / b; % = Jinv(2,2) * dN_deta
757
758     % Matriz B cinemática (3 x 8)
759     % eps = [exx, eyy, gxy]' = B * u
760     B = zeros(3, 8);
761     for i = 1:4
762         B(1, 2*i-1) = dN_dx(i); % exx = dUx/dx
763         B(2, 2*i) = dN_dy(i); % eyy = dUy/dy
764         B(3, 2*i-1) = dN_dy(i); % gxy = dUx/dy + dUy/dx
765         B(3, 2*i) = dN_dx(i);
766     end
767
768     % Contribución al punto de integración (peso = 1 * 1 = 1)
769     K = K + t * (B' * D * B) * detJ;
770 end
771
772 end
773
774 %% FUNCIÓN: mindlinBendingQ4
775 % Matriz de rigidez de flexión de placa para elemento Mindlin Q4 rectangular
776 % con integración selectiva reducida para evitar shear locking.
777 %
778 % INPUTS:
779 % D_f [3x3] Matriz constitutiva de flexión: E*h^3/12/(1-nu^2)*[...]
780 % D_c [2x2] Matriz constitutiva de cortante transverso: kappa*h*G*I
781 % a [m] Semi-dimensión en dirección local x (ancho total = 2a)

```

```

782 % b [m] Semi-dimensión en dirección local y (largo total = 2b)
783 %
784 % OUTPUT:
785 % K [12x12] Matriz de rigidez de flexión en coordenadas locales
786 % DOF: [w1,Rx1,Ry1, w2,Rx2,Ry2, w3,Rx3,Ry3, w4,Rx4,Ry4]
787 % Nodos: (-, -), (+, -), (+, +), (-, +)
788 %
789 % En el sistema global de 6 DOF/nodo:
790 % w -> DOF 3 (Uz)
791 % Rx -> DOF 4 (rotación en X)
792 % Ry -> DOF 5 (rotación en Y)
793 %
794 % Las losas son horizontales (plano XY global), por lo que las
795 % coordenadas locales coinciden con las globales: no se requiere
796 % transformación de ejes.
797 %
798 % Formulación (Ferreira, 2009, Cap. 12, Ec. 12.19-12.24):
799 %
800 % Desplazamientos interpolados:
801 %  $w = \sum_i N_i * w_i$ 
802 %  $\theta_x = \sum_i N_i * R_{x_i}$ 
803 %  $\theta_y = \sum_i N_i * R_{y_i}$ 
804 %
805 % Deformaciones de flexión (Ec. 12.7-12.9):
806 %  $\chi_{xx} = d(\theta_x)/dx$ 
807 %  $\chi_{yy} = d(\theta_y)/dy$ 
808 %  $\chi_{xy} = d(\theta_y)/dx + d(\theta_x)/dy$ 
809 %
810 % Deformaciones de cortante transverso (Ec. 12.10-12.11):
811 %  $\gamma_{xz} = dw/dx + \theta_x$ 
812 %  $\gamma_{yz} = dw/dy + \theta_y$ 
813 %
814 % Rigidez total (Ec. 12.23):
815 %  $K = \int (B_f' * D_f * B_f dA)$  [flexión, 2x2 Gauss]
816 % +  $\int (B_c' * D_c * B_c dA)$  [cortante, 1x1 Gauss reducida]
817
818 function K = mindlinBendingQ4(D_f, D_c, a, b)
819
820 K = zeros(12, 12);
821
822 % -----
823 % PARTE 1: Rigidez de FLEXIÓN - integración 2x2 Gauss (completa)
824 % -----
825 gp = 1.0 / sqrt(3);
826 xi_2 = [-gp, gp, gp, -gp];
827 eta_2 = [-gp, -gp, gp, gp];
828 % Pesos de cuadratura 2x2: todos = 1*1 = 1
829
830 for q = 1:4
831     xi = xi_2(q);
832     eta = eta_2(q);
833
834     % Funciones de forma bilineales Q4
835     % N1=(1-xi)(1-eta)/4, N2=(1+xi)(1-eta)/4
836     % N3=(1+xi)(1+eta)/4, N4=(1-xi)(1+eta)/4

```

```

837 N = [(1-xi)*(1-eta), (1+xi)*(1-eta), ...
838         (1+xi)*(1+eta), (1-xi)*(1+eta)] / 4;
839
840 % Derivadas respecto a coordenadas naturales
841 dN_dxi = [-(1-eta), (1-eta), (1+eta), -(1+eta)] / 4;
842 dN_deta = [-(1-xi), -(1+xi), (1+xi), (1-xi)] / 4;
843
844 % Jacobiano rectangular: det(J) = a*b
845 detJ = a * b;
846 dN_dx = dN_dxi / a;
847 dN_dy = dN_deta / b;
848
849 % Matriz Bf de deformaciones de flexión (3 x 12)
850 % Columnas: [w1, Rx1, Ry1, w2, Rx2, Ry2, w3, Rx3, Ry3, w4, Rx4, Ry4]
851 % Deformaciones: [chi_xx, chi_yy, chi_xy]
852 % chi_xx = d(Rx)/dx -> columna Rx_i: dN_dx(i)
853 % chi_yy = d(Ry)/dy -> columna Ry_i: dN_dy(i)
854 % chi_xy = d(Ry)/dx + d(Rx)/dy -> columna Rx_i: dN_dy(i), Ry_i: dN_dx(i)
855 Bf = zeros(3, 12);
856 for i = 1:4
857     col_w = 3*i - 2; % columna de w_i
858     col_Rx = 3*i - 1; % columna de Rx_i
859     col_Ry = 3*i; % columna de Ry_i
860
861     Bf(1, col_Rx) = dN_dx(i); % chi_xx = d(Rx)/dx
862     Bf(2, col_Ry) = dN_dy(i); % chi_yy = d(Ry)/dy
863     Bf(3, col_Rx) = dN_dy(i); % chi_xy: contribución de Rx
864     Bf(3, col_Ry) = dN_dx(i); % chi_xy: contribución de Ry
865 end
866
867 % Contribución de flexión (peso = 1)
868 K = K + (Bf' * D_f * Bf) * detJ;
869 end
870
871 % -----
872 % PARTE 2: Rigidez de CORTANTE TRANSVERSO - integración 2x2 Gauss
873 % -----
874 % Se usa integración COMPLETA 2x2 (no reducida) para evitar modos
875 % de energía cero (hourglass modes) que aparecen con integración 1x1.
876 %
877 % La integración reducida 1x1 está recomendada en Ferreira (2009) para
878 % placas muy delgadas (a/t >> 50) donde el shear locking es severo.
879 % Para losas de edificio con a/t = a_shell/t_shell = 10, la integración
880 % completa es preferible: evita modos espurios y el error por locking
881 % es despreciable a esta relación de espesor.
882 %
883 % Verificación numérica: rango(K_bend) = 9 con 2x2 (correcto: 12-3=9
884 % modos activos y 3 modos de cuerpo rígido de la placa libre).
885
886 for q = 1:4
887     xi = xi_2(q);
888     eta = eta_2(q);
889
890     N = [(1-xi)*(1-eta), (1+xi)*(1-eta), ...
891           (1+xi)*(1+eta), (1-xi)*(1+eta)] / 4;

```

```

892     dN_dxi = [-(1-eta), (1-eta), (1+eta), -(1+eta)] / 4;
893     dN_deta = [-(1-xi), -(1+xi), (1+xi), (1-xi) ] / 4;
894
895
896     detJ = a * b;
897     dN_dx = dN_dxi / a;
898     dN_dy = dN_deta / b;
899
900     % Matriz Bc de deformaciones de cortante transverso (2 x 12)
901     % Deformaciones: [gamma_xz, gamma_yz]
902     % gamma_xz = dw/dx + Rx
903     % gamma_yz = dw/dy + Ry
904     Bc = zeros(2, 12);
905     for i = 1:4
906         col_w = 3*i - 2;
907         col_Rx = 3*i - 1;
908         col_Ry = 3*i;
909
910         Bc(1, col_w) = dN_dx(i); % gamma_xz: dw/dx
911         Bc(1, col_Rx) = N(i); % gamma_xz: Rx
912         Bc(2, col_w) = dN_dy(i); % gamma_yz: dw/dy
913         Bc(2, col_Ry) = N(i); % gamma_yz: Ry
914     end
915
916     % Contribución de cortante (peso Gauss = 1)
917     K = K + (Bc' * D_c * Bc) * detJ;
918 end
919
920 end

```

B.3. analisis_estatico.m

```

1 %% Análisis estático con carga puntual
2 % Aplica fuerza en azotea y calcula desplazamientos
3
4 %% 1. Verificar scripts previos
5
6 clc
7
8 % 1.1 Verificar que matrices K y M existan
9 if ~exist('Kglobal', 'var') || ~exist('Mglobal', 'var') || ~exist('K_red', 'var'
   )
10     fprintf('Ejecutando scripts previos...\n');
11     run('marco3d_editable.m');
12     run('mrigidez.m');
13     fprintf(' [OK] Scripts cargados\n\n');
14 end
15
16 fprintf(' ANÁLISIS ESTÁTICO - CARGA PUNTUAL\n');
17 fprintf('=====\n\n');
18
19 fprintf('Información del modelo:\n');

```

```

20 fprintf(' Nodos totales: %d\n', nN);
21 fprintf(' GDL totales: %d\n', ngdl);
22 fprintf(' Niveles: %d\n\n', nNiv);
23
24
25 %% 2. Definir nodo de carga puntual
26 % Nodo en azotea - esquina posterior derecha (último nodo)
27 nodo_cargado = (nNiv + 1) * (nCruj + 1) * (nProf + 1);
28
29 fprintf('Nodo seleccionado para carga puntual:\n');
30 fprintf(' Nodo: %d\n', nodo_cargado);
31 fprintf(' Posición: X=%.1f m, Y=%.1f m, Z=%.1f m\n', ...
32         N(nodo_cargado).x, N(nodo_cargado).y, N(nodo_cargado).z);
33 fprintf(' (Esquina posterior-derecha de azotea)\n\n');
34
35
36 %% 3. Verificar condiciones de frontera
37 % 3.1 Los GDL restringidos YA fueron calculados en mrigidez.m
38 if ~exist('gdl_restringidos', 'var') || ~exist('gdl_libres', 'var')
39     error('Variables gdl_restringidos/gdl_libres no encontradas. Ejecuta mrigidez.
40         m primero.');
```

end

```

41
42 fprintf('Condiciones de frontera:\n');
43 fprintf(' GDL restringidos: %d (empotramiento base)\n', length(gdl_restringidos)
44         );
45 fprintf(' GDL libres: %d\n\n', length(gdl_libres));
46
47 %% 4. Definir carga puntual
48
49 fprintf('Definiendo carga puntual...\n');
50
51 % 4.1 Inicializar vector de fuerzas en espacio completo
52 if exist('gdl_mapping', 'var')
53     % 4.1.1 CON DIAFRAGMA: Trabajar en espacio original
54     ngdl_original = gdl_mapping.ngdl_original;
55     F_ext_completo = sparse(ngdl_original, 1);
56     fprintf(' Inicializando fuerzas en espacio completo (%d GDL)\n',
57             ngdl_original);
58 else
59     % 4.1.2 SIN DIAFRAGMA: Usar directamente
60     F_ext_completo = sparse(ngdl, 1);
61 end
62
63 % 4.2 Parámetros de carga (modifica estos según sea necesario)
64 magnitud_X = 100e3; % [N] = 100 kN en dirección X
65 magnitud_Y = 0; % [N] = 0 kN en dirección Y
66 magnitud_Z = 0; % [N] = 0 kN en dirección Z
67
68
69 % 4.3 Aplicar cargas en espacio completo
70
71 if magnitud_X ~= 0
```

```

72     F_ext_completo = cargarEnNodo(F_ext_completo, nodo_cargado, 'X', magnitud_X);
73 end
74
75 if magnitud_Y ~= 0
76     F_ext_completo = cargarEnNodo(F_ext_completo, nodo_cargado, 'Y', magnitud_Y);
77 end
78
79 if magnitud_Z ~= 0
80     F_ext_completo = cargarEnNodo(F_ext_completo, nodo_cargado, 'Z', magnitud_Z);
81 end
82
83 % 4.4 Transformar fuerzas al espacio reducido (por el diafragma)
84
85 if exist('gdl_mapping', 'var')
86     fprintf('\nTransformando fuerzas al espacio con diafragma...\n');
87     % 4.4.1 F_reducido = T' x F_completo
88     F_ext = gdl_mapping.T' * F_ext_completo;
89     fprintf(' F_ext: %d -> %d GDL\n', length(F_ext_completo), length(F_ext));
90     fprintf(' [OK] Fuerzas transformadas\n\n');
91 else
92     F_ext = F_ext_completo;
93 end
94
95 % 4.5 Resumen de carga
96
97 fprintf(' Nodo cargado: %d\n', nodo_cargado);
98 fprintf(' Ubicación: (%.1f, %.1f, %.1f) m\n', ...
99     N(nodo_cargado).x, N(nodo_cargado).y, N(nodo_cargado).z);
100 fprintf(' Fuerza en X: %.0f kN\n', magnitud_X/1000);
101 fprintf(' Fuerza en Y: %.0f kN\n', magnitud_Y/1000);
102 fprintf(' Fuerza en Z: %.0f kN\n\n', magnitud_Z/1000);
103
104
105 %% 5. Resolver sistema  $K \times u = F$ 
106
107
108 fprintf('Resolviendo sistema de ecuaciones...\n');
109
110 % Usar matriz reducida ya calculada en mrigidez.m
111 % K_red = (ya existe)
112
113 % 5.1 Vector de fuerzas reducido
114 F_red = F_ext(gdl_libres);
115
116 % 5.2 Resolver sistema
117 U_red = K_red \ F_red;
118
119 fprintf(' [OK] Sistema resuelto\n');
120 fprintf(' Método: Eliminación directa (backslash operator)\n');
121 fprintf(' Tamaño del sistema: %d x %d\n\n', length(gdl_libres), length(
122     gdl_libres));
123
124 % 5.3 Reconstruir vector completo
125

```

```

126 % Primero expandir en espacio reducido (con diafragma o sin diafragma)
127 U_global = zeros(ngdl, 1);
128 U_global(gdl_libres) = U_red;
129 % U_global(gdl_restringidos) = 0 (ya son ceros)
130
131
132 % 5.4 Reconstruir desplazamientos completos (cuando hay diafragma)
133
134 if exist('gdl_mapping', 'var')
135     fprintf('\nReconstruyendo desplazamientos completos...\n');
136     ngdl_antes = length(U_global); % Guardar tamaño antes
137     % U_completo = T x U_reducido
138     U_completo = gdl_mapping.T * U_global;
139     fprintf(' [OK] Desplazamientos: %d -> %d GDL\n\n', ngdl_antes, length(
        U_completo));
140 else
141     U_completo = U_global;
142 end
143
144
145 %% 6. Análisis de desplazamientos
146 fprintf('Analizando desplazamientos...\n');
147
148 % 6.1 Reorganizar en matriz nN x 6
149 U_nodes = reshape(U_completo, 6, nN)'; % [Ux Uy Uz Rx Ry Rz]
150
151 % 6.2 Magnitud de desplazamiento traslacional
152 U_trans_mag = sqrt(U_nodes(:, 1).^2 + U_nodes(:, 2).^2 + U_nodes(:, 3).^2);
153
154 % 6.3 Nodo con máximo desplazamiento
155 [umax, nodo_umax] = max(U_trans_mag);
156
157 fprintf(' Desplazamiento máximo:\n');
158 fprintf(' Magnitud: %.6f m = %.2f mm\n', umax, umax*1000);
159 fprintf(' Nodo: %d\n', nodo_umax);
160 fprintf(' Ubicación: (%.1f, %.1f, %.1f) m\n\n', ...
161         N(nodo_umax).x, N(nodo_umax).y, N(nodo_umax).z);
162
163 % 6.4 Desplazamiento del nodo cargado
164 fprintf(' Nodo cargado (%d):\n', nodo_cargado);
165 fprintf(' Ux = %.6f m = %.3f mm\n', U_nodes(nodo_cargado, 1), U_nodes(
    nodo_cargado, 1)*1000);
166 fprintf(' Uy = %.6f m = %.3f mm\n', U_nodes(nodo_cargado, 2), U_nodes(
    nodo_cargado, 2)*1000);
167 fprintf(' Uz = %.6f m = %.3f mm\n\n', U_nodes(nodo_cargado, 3), U_nodes(
    nodo_cargado, 3)*1000);
168
169
170 %% 7. Calcular desplazamientos por nivel
171
172 fprintf('Desplazamientos por nivel:\n');
173 fprintf('-----\n');
174 fprintf(' Nivel Altura Ux (mm) Uy (mm) Uz (mm)\n');
175 fprintf('-----\n');
176

```

```

177 % 7.1 Para cada nivel, promediar desplazamientos de todos los nodos
178 for iNiv = 1:(nNiv + 1)
179     % 7.1.1 Altura del nivel
180     z_nivel = (iNiv - 1) * h;
181
182     % 7.1.2 Encontrar nodos en este nivel
183     nodos_nivel = [];
184     for iP = 1:(nProf + 1)
185         for iC = 1:(nCruj + 1)
186             idx = nodeIndex(iNiv, iC, iP, nNiv, nCruj);
187             nodos_nivel = [nodos_nivel, idx];
188         end
189     end
190
191     % 7.1.3 Promediar desplazamientos
192     Ux_prom = mean(U_nodes(nodos_nivel, 1)) * 1000; % mm
193     Uy_prom = mean(U_nodes(nodos_nivel, 2)) * 1000; % mm
194     Uz_prom = mean(U_nodes(nodos_nivel, 3)) * 1000; % mm
195
196     fprintf(' %2d %6.2f %8.3f %8.3f %8.3f\n', ...
197         iNiv-1, z_nivel, Ux_prom, Uy_prom, Uz_prom);
198 end
199 fprintf('-----\n\n');
200
201
202 %% 8. Calcular desplazamientos del entrepiso
203
204
205 fprintf('Desplazamientos del entrepiso:\n');
206 fprintf('-----\n');
207 fprintf(' Piso Drift_X (%%) Drift_Y (%%) Drift_Z (%%)\n');
208 fprintf('-----\n');
209
210 for iPiso = 1:nNiv
211     % 8.1 Niveles inferior y superior del piso
212     iNiv_inf = iPiso; % Base del piso
213     iNiv_sup = iPiso + 1; % Techo del piso
214
215     % 8.2 Nodos en cada nivel
216     nodos_inf = [];
217     nodos_sup = [];
218
219     for iP = 1:(nProf + 1)
220         for iC = 1:(nCruj + 1)
221             idx_inf = nodeIndex(iNiv_inf, iC, iP, nNiv, nCruj);
222             idx_sup = nodeIndex(iNiv_sup, iC, iP, nNiv, nCruj);
223             nodos_inf = [nodos_inf, idx_inf];
224             nodos_sup = [nodos_sup, idx_sup];
225         end
226     end
227
228     % 8.3 Desplazamientos promedio
229     Ux_inf = mean(U_nodes(nodos_inf, 1));
230     Uy_inf = mean(U_nodes(nodos_inf, 2));
231     Uz_inf = mean(U_nodes(nodos_inf, 3));

```

```

232
233   Ux_sup = mean(U_nodes(nodos_sup, 1));
234   Uy_sup = mean(U_nodes(nodos_sup, 2));
235   Uz_sup = mean(U_nodes(nodos_sup, 3));
236
237   % 8.4 Desplazamientos (desplazamiento relativo / altura)
238   drift_X = abs(Ux_sup - Ux_inf) / h * 100; % %
239   drift_Y = abs(Uy_sup - Uy_inf) / h * 100; % %
240   drift_Z = abs(Uz_sup - Uz_inf) / h * 100; % %
241
242   fprintf(' %2d %6.4f %6.4f %6.4f\n', ...
243           iPiso, drift_X, drift_Y, drift_Z);
244 end
245   fprintf('-----\n\n');
246
247
248 %% 9. Calcular fuerzas internas
249 fprintf('Calculando fuerzas internas en elementos...\n');
250
251 FuerzasInternas = struct;
252
253 for e = 1:length(Barras)
254     % 9.1 Nodos extremos
255     ni = Barras(e).nodo1;
256     nj = Barras(e).nodo2;
257
258     % 9.2 Coordenadas
259     xi = N(ni).x; yi = N(ni).y; zi = N(ni).z;
260     xj = N(nj).x; yj = N(nj).y; zj = N(nj).z;
261
262     % 9.3 Longitud (usar la ya calculada)
263     L_elem = Barras(e).L;
264
265     % 9.4 Propiedades según tipo (usar campo .tipo)
266     if strcmp(Barras(e).tipo, 'columna')
267         Ae = A_col;
268         Iy = Iy_col;
269         Iz = Iz_col;
270         Je = J_col;
271     else
272         Ae = A_vig;
273         Iy = Iy_vig;
274         Iz = Iz_vig;
275         Je = J_vig;
276     end
277
278     % 9.5 Rigidez local
279     kLocal = localStiffness3D(E, G, Je, Ae, Iy, Iz, L_elem);
280
281     % 9.6 Transformación
282     T = transformMatrix3D(xi, yi, zi, xj, yj, zj);
283
284     % 9.7 GDL del elemento
285     dofs_i = 6 * (ni - 1) + (1:6);
286     dofs_j = 6 * (nj - 1) + (1:6);

```

```

287     dofs = [dofs_i, dofs_j];
288
289     % 9.8 Desplazamientos del elemento
290     U_elem = U_completo(dofs);
291     u_local = T * U_elem;
292
293     % 9.9 Fuerzas internas
294     f_local = kLocal * u_local;
295
296     % 9.10 Guardar
297     FuerzasInternas(e).nodo1 = ni;
298     FuerzasInternas(e).nodo2 = nj;
299     FuerzasInternas(e).f_local = f_local;
300     FuerzasInternas(e).L = L_elem;
301     FuerzasInternas(e).tipo = Barras(e).tipo;
302 end
303
304 fprintf(' [OK] %d elementos procesados\n\n', length(Barras));
305
306
307 %% 10. Visualización con carga
308 fprintf('Generando visualización...\n');
309
310 % 10.1 Factor de escala para deformada (ajustar según necesites)
311 factor_escala = 100; % Amplifica desplazamientos para visualizar
312
313 % 10.2 Colores
314 colorOriginal = [0.13, 0.16, 0.36]; % Azul oscuro
315 colorDeformada = [0.99, 0.62, 0.01]; % Naranja
316 colorNodoCargado = [0.94, 0.14, 0.24]; % Rojo
317 colorFlecha = [1.0, 0.0, 0.0]; % Rojo brillante para flecha
318
319 % 10.3 Crear figura
320 figure('Name', 'Análisis Estático - Deformada', ...
321     'NumberTitle', 'off', ...
322     'Position', [50, 50, 1200, 800]);
323
324 grid on;
325 axis equal;
326 view(40, 20);
327 xlabel('X [m]', 'FontSize', 12);
328 ylabel('Y [m]', 'FontSize', 12);
329 zlabel('Z [m]', 'FontSize', 12);
330 title(sprintf('Marco 3D - Deformada (Factor escala: %dx)', factor_escala), ...
331     'FontSize', 14, 'FontWeight', 'bold');
332
333 hold on;
334
335 % 10.4 Dibujar estructura original
336 h_original = []; % Inicializar handle
337 for e = 1:length(Barras)
338     ni = Barras(e).nodo1;
339     nj = Barras(e).nodo2;
340
341     h_temp = plot3([N(ni).x, N(nj).x], ...

```

```

342         [N(ni).y, N(nj).y], ...
343         [N(ni).z, N(nj).z], ...
344         'Color', colorOriginal, ...
345         'LineWidth', 1.5, ...
346         'LineStyle', '-');
347
348     if isempty(h_original)
349         h_original = h_temp; % Guardar solo el primero para leyenda
350     end
351 end
352
353 % 10.5 Dibujar estructura deformada
354 h_deformada = []; % Almacenar TODOS los handles
355 for e = 1:length(Barras)
356     ni = Barras(e).nodo1;
357     nj = Barras(e).nodo2;
358
359     % 10.5.1 Coordenadas deformadas
360     xi_def = N(ni).x + U_nodes(ni, 1) * factor_escala;
361     yi_def = N(ni).y + U_nodes(ni, 2) * factor_escala;
362     zi_def = N(ni).z + U_nodes(ni, 3) * factor_escala;
363
364     xj_def = N(nj).x + U_nodes(nj, 1) * factor_escala;
365     yj_def = N(nj).y + U_nodes(nj, 2) * factor_escala;
366     zj_def = N(nj).z + U_nodes(nj, 3) * factor_escala;
367
368     h_temp = plot3([xi_def, xj_def], ...
369                 [yi_def, yj_def], ...
370                 [zi_def, zj_def], ...
371                 'Color', colorDeformada, ...
372                 'LineWidth', 2.0, ...
373                 'LineStyle', '-');
374
375     h_deformada = [h_deformada; h_temp]; % Guardar TODOS los handles
376 end
377
378 % 10.6 Marcar nodo cargado
379 h_nodo = plot3(N(nodo_cargado).x, N(nodo_cargado).y, N(nodo_cargado).z, ...
380             'o', 'MarkerSize', 15, ...
381             'MarkerFaceColor', colorNodoCargado, ...
382             'MarkerEdgeColor', 'k', ...
383             'LineWidth', 2);
384
385
386 % 10.7 Dibujar flecha
387
388
389 % 10.7.1 Coordenadas del nodo cargado
390 x_nodo = N(nodo_cargado).x;
391 y_nodo = N(nodo_cargado).y;
392 z_nodo = N(nodo_cargado).z;
393
394 % 10.7.2 Calcular escala automática de flecha (proporcional al tamaño del
395 edificio)
396 dim_edificio = max([nProf * w, nCruj * L, nNiv * h]);

```

```

396 escala_flecha = dim_edificio * 0.15; % 15% del tamaño del edificio
397
398 % 10.7.3 Componentes del vector de carga (normalizar)
399 F_total = sqrt(magnitud_X^2 + magnitud_Y^2 + magnitud_Z^2);
400
401 if F_total > 0
402     % Dirección normalizada
403     dir_x = magnitud_X / F_total;
404     dir_y = magnitud_Y / F_total;
405     dir_z = magnitud_Z / F_total;
406
407     % Punto final de la flecha
408     x_flecha = x_nodo + dir_x * escala_flecha;
409     y_flecha = y_nodo + dir_y * escala_flecha;
410     z_flecha = z_nodo + dir_z * escala_flecha;
411
412     % Dibujar flecha con quiver3
413     h_flecha = quiver3(x_nodo, y_nodo, z_nodo, ...
414                       dir_x * escala_flecha, ...
415                       dir_y * escala_flecha, ...
416                       dir_z * escala_flecha, ...
417                       0, ... % AutoScale = 0 (usar longitud exacta)
418                       'Color', colorFlecha, ...
419                       'LineWidth', 4, ...
420                       'MaxHeadSize', 0.8);
421
422     % Etiqueta de la fuerza
423     % Posición del texto (un poco más allá de la punta de la flecha)
424     x_texto = x_flecha + dir_x * escala_flecha * 0.3;
425     y_texto = y_flecha + dir_y * escala_flecha * 0.3;
426     z_texto = z_flecha + dir_z * escala_flecha * 0.3;
427
428     % Texto con magnitud
429     if magnitud_X ~= 0 && magnitud_Y == 0 && magnitud_Z == 0
430         texto_carga = sprintf('Fx = %.0f kN', magnitud_X/1000);
431     elseif magnitud_Y ~= 0 && magnitud_X == 0 && magnitud_Z == 0
432         texto_carga = sprintf('Fy = %.0f kN', magnitud_Y/1000);
433     elseif magnitud_Z ~= 0 && magnitud_X == 0 && magnitud_Y == 0
434         texto_carga = sprintf('Fz = %.0f kN', magnitud_Z/1000);
435     else
436         texto_carga = sprintf('F = %.0f kN', F_total/1000);
437     end
438
439     text(x_texto, y_texto, z_texto, texto_carga, ...
440         'FontSize', 13, ...
441         'FontWeight', 'bold', ...
442         'Color', colorFlecha, ...
443         'BackgroundColor', [1, 1, 1, 0.8], ...
444         'EdgeColor', colorFlecha, ...
445         'LineWidth', 1.5, ...
446         'Margin', 3, ...
447         'HorizontalAlignment', 'center');
448 end
449
450 % 10.8 Leyenda (usar solo el primer handle)

```

```

451 if exist('h_flecha', 'var')
452     legend([h_original, h_deformada(1), h_nodo, h_flecha], ...
453           {'Original', 'Deformada', 'Nodo cargado', 'Carga aplicada'}, ...
454           'Location', 'best', ...
455           'FontSize', 11);
456 else
457     legend([h_original, h_deformada(1), h_nodo], ...
458           {'Original', 'Deformada', 'Nodo cargado'}, ...
459           'Location', 'best', ...
460           'FontSize', 11);
461 end
462
463 % 10.9 Control interactivo de amplificación con slider
464
465 % Crear panel de control
466 panel_control = uipanel('Title', 'Control de Amplificación', ...
467                        'FontSize', 10, ...
468                        'FontWeight', 'bold', ...
469                        'BackgroundColor', [0.94, 0.94, 0.94], ...
470                        'Position', [0.02, 0.85, 0.15, 0.12]);
471
472 % Texto indicador del factor actual
473 txt_factor = uicontrol('Parent', panel_control, ...
474                      'Style', 'text', ...
475                      'String', sprintf('Factor: %dx', factor_escalas), ...
476                      'FontSize', 11, ...
477                      'FontWeight', 'bold', ...
478                      'BackgroundColor', [0.94, 0.94, 0.94], ...
479                      'Position', [10, 45, 120, 25]);
480
481 % Slider para ajustar amplificación (rango de 1 a 500)
482 slider_amp = uicontrol('Parent', panel_control, ...
483                      'Style', 'slider', ...
484                      'Min', 1, ...
485                      'Max', 500, ...
486                      'Value', factor_escalas, ...
487                      'Position', [10, 25, 120, 20], ...
488                      'Callback', @(src, evt) actualizarAmplificacion(src,
489                                h_deformada, txt_factor));
489
490 % Etiquetas de rango
491 uicontrol('Parent', panel_control, ...
492          'Style', 'text', ...
493          'String', '1x', ...
494          'FontSize', 8, ...
495          'BackgroundColor', [0.94, 0.94, 0.94], ...
496          'Position', [10, 5, 20, 15]);
497
498 uicontrol('Parent', panel_control, ...
499          'Style', 'text', ...
500          'String', '500x', ...
501          'FontSize', 8, ...
502          'BackgroundColor', [0.94, 0.94, 0.94], ...
503          'Position', [100, 5, 30, 15]);
504

```

```

505 % Guardar datos necesarios en UserData del slider
506 slider_amp.UserData = struct('U_nodes', U_nodes, ...
507     'N', N, ...
508     'Barras', Barras);
509
510 hold off;
511
512 fprintf(' [OK] Visualización generada\n\n');
513
514
515 %% 11. Resumen final
516 fprintf(' RESUMEN DE ANÁLISIS ESTÁTICO\n');
517 fprintf('=====\n');
518 fprintf('Carga aplicada:\n');
519 fprintf(' Nodo: %d (azotea)\n', nodo_cargado);
520 fprintf(' Fx: %.0f kN\n', magnitud_X/1000);
521 fprintf(' Fy: %.0f kN\n', magnitud_Y/1000);
522 fprintf(' Fz: %.0f kN\n\n', magnitud_Z/1000);
523
524 fprintf('Desplazamiento máximo:\n');
525 fprintf(' Magnitud: %.3f mm\n', umax*1000);
526 fprintf(' Nodo: %d\n', nodo_umax);
527
528 fprintf('Desplazamiento en nodo cargado:\n');
529 fprintf(' Ux: %.3f mm\n', U_nodes(nodo_cargado, 1)*1000);
530 fprintf(' Uy: %.3f mm\n', U_nodes(nodo_cargado, 2)*1000);
531 fprintf(' Uz: %.3f mm\n\n', U_nodes(nodo_cargado, 3)*1000);
532
533 % 11.1 Drift máximo
534 drifts_X = [];
535 for iPiso = 1:nNiv
536     iNiv_inf = iPiso;
537     iNiv_sup = iPiso + 1;
538     nodos_inf = [];
539     nodos_sup = [];
540     for iP = 1:(nProf + 1)
541         for iC = 1:(nCruj + 1)
542             nodos_inf = [nodos_inf, nodeIndex(iNiv_inf, iC, iP, nNiv, nCruj)];
543             nodos_sup = [nodos_sup, nodeIndex(iNiv_sup, iC, iP, nNiv, nCruj)];
544         end
545     end
546     drift = abs(mean(U_nodes(nodos_sup, 1)) - mean(U_nodes(nodos_inf, 1))) / h *
547         100;
548     drifts_X = [drifts_X, drift];
549 end
550 [drift_max, piso_max] = max(drifts_X);
551
552 fprintf('Drift máximo:\n');
553 fprintf(' Dirección X: %.4f %% (piso %d)\n', drift_max, piso_max);
554 fprintf('=====\n\n');
555
556 fprintf('[OK] Análisis estático completado\n\n');
557
558 %% Funciones locales

```

```

559 % Función que carga en nodo para agregar fuerza/momento en un nodo específico
560 function F = cargarEnNodo(F, nodo, direccion, magnitud)
561     % Agrega fuerza/momento en un nodo específico
562     switch upper(direccion)
563         case 'X', idx = 1;
564         case 'Y', idx = 2;
565         case 'Z', idx = 3;
566         case 'RX', idx = 4;
567         case 'RY', idx = 5;
568         case 'RZ', idx = 6;
569         otherwise
570             error('Dirección inválida. Usa X, Y, Z, RX, RY o RZ.');
```

```

571     end
572     dof = 6 * (nodo - 1) + idx;
573     F(dof) = F(dof) + magnitud;
574 end
575
576 % Función que calcula el índice del nodo en la malla 3D
577 function idx = nodeIndex(iN, iC, iP, nNiv, nCruj)
578     % Calcula índice de nodo en malla 3D
579     offset_profundidad = (iP - 1) * (nCruj + 1) * (nNiv + 1);
580     offset_crujia = (iC - 1) * (nNiv + 1);
581     offset_nivel = iN;
582     idx = offset_profundidad + offset_crujia + offset_nivel;
583 end
584
585 % Función matriz de rigidez local 12x12 para elemento viga-columna 3D
586 function k = localStiffness3D(E, G, J, A, Iy, Iz, L)
587     k = zeros(12, 12);
588
589     % Coeficientes
590     EA_L = E * A / L;
591     GJ_L = G * J / L;
592     EIy_L3 = E * Iy / L^3;
593     EIZ_L3 = E * Iz / L^3;
594     EIy_L2 = E * Iy / L^2;
595     EIZ_L2 = E * Iz / L^2;
596     EIy_L = E * Iy / L;
597     EIZ_L = E * Iz / L;
598
599     % RIGIDEZ AXIAL (GDL 1 y 7: u)
600     k(1, 1) = EA_L;
601     k(1, 7) = -EA_L;
602     k(7, 1) = -EA_L;
603     k(7, 7) = EA_L;
604
605     % RIGIDEZ TORSIONAL (GDL 4 y 10: thetax)
606     k(4, 4) = GJ_L;
607     k(4, 10) = -GJ_L;
608     k(10, 4) = -GJ_L;
609     k(10, 10) = GJ_L;
610
611     % FLEXIÓN EN PLANO XZ (GDL 2, 6, 8, 12: v, thetaz)
612     k(2, 2) = 12 * EIz_L3;
613     k(2, 6) = 6 * EIz_L2;
```

```

614 k(2, 8) = -12 * EIz_L3;
615 k(2, 12) = 6 * EIz_L2;
616
617 k(6, 2) = 6 * EIz_L2;
618 k(6, 6) = 4 * EIz_L;
619 k(6, 8) = -6 * EIz_L2;
620 k(6, 12) = 2 * EIz_L;
621
622 k(8, 2) = -12 * EIz_L3;
623 k(8, 6) = -6 * EIz_L2;
624 k(8, 8) = 12 * EIz_L3;
625 k(8, 12) = -6 * EIz_L2;
626
627 k(12, 2) = 6 * EIz_L2;
628 k(12, 6) = 2 * EIz_L;
629 k(12, 8) = -6 * EIz_L2;
630 k(12, 12) = 4 * EIz_L;
631
632 % FLEXIÓN EN PLANO XY (GDL 3, 5, 9, 11: w, thetay)
633 k(3, 3) = 12 * EIy_L3;
634 k(3, 5) = -6 * EIy_L2;
635 k(3, 9) = -12 * EIy_L3;
636 k(3, 11) = -6 * EIy_L2;
637
638 k(5, 3) = -6 * EIy_L2;
639 k(5, 5) = 4 * EIy_L;
640 k(5, 9) = 6 * EIy_L2;
641 k(5, 11) = 2 * EIy_L;
642
643 k(9, 3) = -12 * EIy_L3;
644 k(9, 5) = 6 * EIy_L2;
645 k(9, 9) = 12 * EIy_L3;
646 k(9, 11) = 6 * EIy_L2;
647
648 k(11, 3) = -6 * EIy_L2;
649 k(11, 5) = 2 * EIy_L;
650 k(11, 9) = 6 * EIy_L2;
651 k(11, 11) = 4 * EIy_L;
652 end
653
654 % Función para la transformación de coordenadas locales a globales
655 function T = transformMatrix3D(xi, yi, zi, xj, yj, zj)
656
657 % Vector del elemento
658 dx = xj - xi;
659 dy = yj - yi;
660 dz = zj - zi;
661 L = sqrt(dx^2 + dy^2 + dz^2);
662
663 % Cosenos directores del eje local x
664 cx = dx / L;
665 cy = dy / L;
666 cz = dz / L;
667
668 % Determinar ejes locales y, z

```

```

669     tol = 1e-6;
670
671     if abs(cx) < tol && abs(cy) < tol
672         % ELEMENTO VERTICAL (columna)
673         vy = [1; 0; 0];
674     else
675         % ELEMENTO HORIZONTAL O INCLINADO
676         vz_ref = [0; 0; 1];
677         vx = [cx; cy; cz];
678         vy = cross(vz_ref, vx);
679         vy = vy / norm(vy);
680     end
681
682     % Eje local z = x x y
683     vx = [cx; cy; cz];
684     vz = cross(vx, vy);
685     vz = vz / norm(vz);
686
687     % Matriz de rotación 3x3
688     R = [vx, vy, vz]';
689
690     % Matriz de transformación 12x12
691     T = zeros(12, 12);
692     T(1:3, 1:3) = R;
693     T(4:6, 4:6) = R;
694     T(7:9, 7:9) = R;
695     T(10:12, 10:12) = R;
696 end
697
698 % Función para actualizar amplificación dinámicamente
699 function actualizarAmplificacion(slider, handles_deformada, txt_factor)
700     % Obtener nuevo factor de escala
701     nuevo_factor = round(slider.Value);
702
703     % Actualizar texto del panel
704     set(txt_factor, 'String', sprintf('Factor: %dx', nuevo_factor));
705
706     % Actualizar título de la figura
707     title(sprintf('Marco 3D - Deformada (Factor escala: %dx)', nuevo_factor), ...
708           'FontSize', 14, 'FontWeight', 'bold');
709
710     % Recuperar datos guardados
711     U_nodos = slider.UserData.U_nodos;
712     N = slider.UserData.N;
713     Barras = slider.UserData.Barras;
714
715     % Actualizar cada línea deformada
716     for e = 1:length(handles_deformada)
717         if isvalid(handles_deformada(e))
718             % Nodos del elemento
719             ni = Barras(e).nodo1;
720             nj = Barras(e).nodo2;
721
722             % Nuevas coordenadas deformadas con nuevo factor
723             xi_def = N(ni).x + U_nodos(ni, 1) * nuevo_factor;

```

```

724     yi_def = N(ni).y + U_nodes(ni, 2) * nuevo_factor;
725     zi_def = N(ni).z + U_nodes(ni, 3) * nuevo_factor;
726
727     xj_def = N(nj).x + U_nodes(nj, 1) * nuevo_factor;
728     yj_def = N(nj).y + U_nodes(nj, 2) * nuevo_factor;
729     zj_def = N(nj).z + U_nodes(nj, 3) * nuevo_factor;
730
731     % Actualizar coordenadas de la línea
732     set(handles_deformada(e), 'XData', [xi_def, xj_def], ...
733         'YData', [yi_def, yj_def], ...
734         'ZData', [zi_def, zj_def]);
735
736     end
737
738     % Forzar actualización de la figura
739     drawnow;
740 end

```

B.4. analisis_dinamico.m

```

1  %% Análisis dinámico modal y respuesta espectral desacoplada
2  % Verificar si las variables ya existen
3  if ~exist('Kglobal', 'var') || ~exist('Mglobal', 'var')
4      fprintf('Ejecutando scripts previos...\n');
5      run('marco3d_editable.m');
6      run('mrigidez.m');
7  end
8
9  %% 1. Verificación e inicialización
10 if ~exist('ngdl', 'var')
11     error('Variable ngdl no encontrada. Ejecuta scripts previos.');
```

```

12 end
13
14 fprintf(' ANÁLISIS DINÁMICO MODAL\n');
15 fprintf('Nodos: %d\n', nN);
16 fprintf('GDL: %d\n\n', ngdl);
17
18 %% 2. Usar matriz de masa previamente calculada
19 % Mglobal ya fue calculada en marco3d_editable.m
20 % Contiene: masa de columnas + vigas + losas
21
22 fprintf('Matriz de masa:\n');
23 fprintf(' Tamaño: %d -- %d\n', size(Mglobal, 1), size(Mglobal, 2));
24 fprintf(' Masa total: %.0f ton\n\n', sum(masa_nodal)/1000);
25
26 %% 3. Usar GDL ya calculados
27 % gdl_libres ya fue calculado en mrigidez.m
28
29 if ~exist('gdl_libres', 'var')
30     error('Variable gdl_libres no encontrada. Ejecuta mrigidez.m primero.');
```

```

31 end
32

```

```

33 fprintf('Grados de libertad:\n');
34 fprintf(' Total: %d\n', ngdl);
35 fprintf(' Restringidos: %d\n', length(gdl_restringidos));
36 fprintf(' Libres: %d\n\n', length(gdl_libres));
37
38 %% 4. Usar matrices reducidas ya calculadas
39 % K_red y M_red ya fueron calculadas en mrigidez.m
40
41 if ~exist('K_red', 'var') || ~exist('M_red', 'var')
42     error('Matrices reducidas no encontradas. Ejecuta mrigidez.m primero.');
```

end

```

44
45 fprintf('Matrices reducidas:\n');
46 fprintf(' K_red: %d - %d\n', size(K_red, 1), size(K_red, 2));
47 fprintf(' M_red: %d - %d\n\n', size(M_red, 1), size(M_red, 2));
48
49 %% 5. Análisis Modal
50
51 fprintf('Resolviendo problema de autovalores...\n');
```

52

```

53 % 5.1 Número de modos a calcular
54 n_modos = min(12, length(gdl_libres)); % Máximo 12 o todos si hay menos
55
56 % 5.2 Problema de autovalores generalizado
57 [phi, lambda] = eigs(K_red, M_red, n_modos, 'smallestabs');
```

58

```

59 % 5.3 Ordenar por frecuencia ascendente
60 [lambda_sorted, idx] = sort(diag(lambda));
61 phi = phi(:, idx);
62 lambda = diag(lambda_sorted);
63
64 % 5.4 Frecuencias y períodos
65 omega = sqrt(lambda_sorted); % [rad/s]
66 frecuencias = omega / (2*pi); % [Hz]
67 periodos = 1 ./ frecuencias; % [s]
68
69 fprintf('%d modos calculados\n\n', n_modos);
70
71 % 5.5 Mostrar resultados
72 fprintf(' FRECUENCIAS Y PERÍODOS NATURALES\n');
73 fprintf(' Modo Frecuencia (Hz) Período (s)\n');
```

74

```

75 for i = 1:min(6, n_modos)
76     fprintf(' %2d %8.4f %8.4f\n', i, frecuencias(i), periodos(i));
77 end
78 fprintf('...\n');
```

79

```

80 %% 6. Normalización modal
81
82 fprintf('Normalizando modos respecto a masa...\n');
```

83

```

84 for i = 1:size(phi, 2)
85     phi(:, i) = phi(:, i) / sqrt(phi(:, i)' * M_red * phi(:, i));
86 end
87
```

```

88 fprintf('Modos normalizados\n\n');
89
90 % 6.1 Verificar ortogonalidad
91 fprintf('Verificando ortogonalidad de modos...\n');
92
93 % 6.2 Matriz de masa modal (debe ser de identidad)
94 M_modal = phi' * M_red * phi;
95 error_ortog_M = norm(M_modal - eye(size(M_modal)), 'fro');
96
97 % 6.3 Matriz de rigidez modal (debe ser diagonal con lambda)
98 K_modal = phi' * K_red * phi;
99 error_ortog_K = norm(K_modal - diag(diag(K_modal)), 'fro') / norm(K_modal, 'fro'
    );
100
101 fprintf(' Error ortogonalidad masa: %.2e (debe ser < 1e-10)\n', error_ortog_M);
102 fprintf(' Error ortogonalidad rigidez: %.2e (debe ser < 1e-6)\n\n',
    error_ortog_K);
103
104 if error_ortog_M > 1e-10
105     warning('Modos no son ortogonales respecto a masa (error > 1e-10). Revisar
    matrices.');
```

106 **end**

```

107
108 %% 7. Factores de participación modal
109
110 fprintf('Calculando factores de participación...\n');
111
112 % 7.1 Vectores de influencia (1 en GDL de traslación, 0 en rotación)
113 n_gdl_libres = length(gdl_libres);
114
115 rx = zeros(n_gdl_libres, 1);
116 ry = zeros(n_gdl_libres, 1);
117 rz = zeros(n_gdl_libres, 1);
118
119 for i = 1:n_gdl_libres
120     % Recuperar índice en el espacio original (6*nN) para identificar
121     % correctamente el tipo de GDL, independientemente de si el espacio
122     % fue reducido por condensación de diafragma o no.
123     if exist('gdl_mapping', 'var')
124         % CON DIAFRAGMA: gdl_libres(i) es un índice en el espacio reducido.
125         % gdl_mapping.independientes mapea ese índice al espacio original.
126         idx_original = gdl_mapping.independientes(gdl_libres(i));
127     else
128         % SIN DIAFRAGMA: gdl_libres(i) ya es un índice en el espacio original.
129         idx_original = gdl_libres(i);
130     end
131     tipo_gdl = mod(idx_original - 1, 6) + 1; % 1=Ux, 2=Uy, 3=Uz, 4=Rx, 5=Ry, 6=Rz
132
133     if tipo_gdl == 1 % Ux
134         rx(i) = 1;
135     elseif tipo_gdl == 2 % Uy
136         ry(i) = 1;
137     elseif tipo_gdl == 3 % Uz
138         rz(i) = 1;
139     end

```

```

140 % Rotaciones (tipo 4, 5, 6) quedan en 0
141 end
142
143 % 7.2 Factores de participación modal
144 Gamma_X = phi' * M_red * rx;
145 Gamma_Y = phi' * M_red * ry;
146 Gamma_Z = phi' * M_red * rz;
147
148 fprintf('Factores calculados\n\n');
149
150 % 7.3 Masas modales efectivas
151 M_eff_X = Gamma_X.^2;
152 M_eff_Y = Gamma_Y.^2;
153 M_eff_Z = Gamma_Z.^2;
154
155 % 7.4 Masa total movilizable por dirección (forma cuadrática)
156 % La expresión correcta es r'*M_red*r, no la suma de la diagonal de una
157 % submatriz. Esto es necesario porque M_red = T'*M*T es densa en general:
158 % la condensación del diafragma introduce acoplamientos de inercia fuera
159 % de la diagonal. La suma de participaciones de todos los modos debe
160 % converger a este valor (NTC-Sismo 2020, ec. 6.1.1).
161 M_total_X = full(rx' * M_red * rx);
162 M_total_Y = full(ry' * M_red * ry);
163 M_total_Z = full(rz' * M_red * rz);
164
165 % 7.5 Porcentaje de participación (masa por dirección)
166 part_X = M_eff_X / M_total_X * 100;
167 part_Y = M_eff_Y / M_total_Y * 100;
168 part_Z = M_eff_Z / M_total_Z * 100;
169
170 fprintf(' MASAS MODALES PARTICIPANTES\n');
171 fprintf(' Modo X (%%) Y (%%) Z (%%)\n');
172 for i = 1:min(6, n_modos)
173     fprintf(' %2d %6.2f %6.2f %6.2f\n', ...
174           i, part_X(i), part_Y(i), part_Z(i));
175 end
176
177 fprintf(' Suma %6.2f %6.2f %6.2f\n', ...
178       sum(part_X(1:min(6,n_modos))), ...
179       sum(part_Y(1:min(6,n_modos))), ...
180       sum(part_Z(1:min(6,n_modos))));
181
182
183 %% 8. Análisis espectral - Espectro de SASID
184
185 fprintf('Generando espectro de diseño (SASID)...\n');
186
187 % 8.1 CARGAR ESPECTRO DEL SASID
188 % Cargar archivo Q2 (espectro del SASID ya reducido por Q=2)
189 archivo_espectro = 'Q2.txt';
190
191 if ~exist(archivo_espectro, 'file')
192     error('Archivo de espectro no encontrado: %s', archivo_espectro);
193 end
194

```

```

195 data_espectro = load(archivo_espectro);
196 T_sasid = data_espectro(:, 1); % Períodos [s]
197 Sa_sasid_gal = data_espectro(:, 2); % Sa [Gal = cm/s^2]
198
199 % 8.2 Convertir Gal a g
200 Sa_sasid = Sa_sasid_gal / 981; % 1 g = 981 Gal
201
202 % 8.3 Configuración de factor de Q
203 Q_sasid = 2.0; % Factor Q del espectro SASID
204 Q_adicional = 1.0; % Factor adicional (1.0 = no reducir más, 2.0 = reducir a la
    mitad)
205 Q_efectivo = Q_sasid * Q_adicional;
206
207
208 fprintf(' Espectro cargado: %s\n', archivo_espectro);
209 fprintf(' Puntos: %d\n', length(T_sasid));
210 fprintf(' Rango períodos: %.2f - %.2f s\n', min(T_sasid), max(T_sasid));
211 fprintf(' Sa máximo: %.2f g (@ Q=%g)\n', max(Sa_sasid), Q_sasid);
212 fprintf(' Q base (SASID): %.1f\n', Q_sasid);
213 fprintf(' Q adicional: %.1f\n', Q_adicional);
214 fprintf(' Q efectivo: %.1f\n\n', Q_efectivo);
215
216 % 8.4 Interpolar espectro en períodos modales
217 Sa = zeros(n_modos, 1);
218
219 for i = 1:n_modos
220     % 8.4.1 Interpolar Sa para el período del modo i
221     Sa_g = interp1(T_sasid, Sa_sasid, periodos(i), 'linear', 'extrap');
222
223     % 8.4.2 Aplicar reducción adicional si se requiere
224     Sa_g = Sa_g / Q_adicional;
225
226     % 8.4.3 Convertir de g a m/s^2
227     Sa(i) = Sa_g * 9.81;
228 end
229
230 fprintf('Valores espectrales en modos principales:\n');
231 for i = 1:min(6, n_modos)
232     fprintf(' Modo %d (T=%.2fs): Sa = %.3f g\n', i, periodos(i), Sa(i)/9.81);
233 end
234 fprintf('\n');
235
236 % 8.5 Factores de participación (calculados en Sección 7)
237 Gamma_X = Gamma_X(:);
238 Gamma_Y = Gamma_Y(:);
239 Gamma_Z = Gamma_Z(:);
240 Sa = Sa(:);
241
242 % 8.6 Desplazamientos modales máximos
243 % NOTA: Sd_X, Sd_Y, Sd_Z son vectores modales ponderados por Gamma.
244 % Umax_modal_X = phi * Sd_X es la superposición ALGEBRAICA de las
245 % contribuciones modales. Este último resultado NO es el desplazamiento máximo
246 % espectral (que requiere combinación SRSS o CQC). Se calcula aquí
247 % únicamente como paso intermedio para la combinación SRSS de la
248 % Sección 8.7 y NO debe usarse directamente como resultado de diseño.

```

```

249
250 fprintf('Calculando desplazamientos espectrales...\n');
251
252 Sd_X = (Sa ./ (omega.^2)) .* Gamma_X;
253 Sd_Y = (Sa ./ (omega.^2)) .* Gamma_Y;
254 Sd_Z = (Sa ./ (omega.^2)) .* Gamma_Z;
255
256 Umax_modal_X = phi * Sd_X;
257 Umax_modal_Y = phi * Sd_Y;
258 Umax_modal_Z = phi * Sd_Z;
259
260 fprintf('Desplazamientos modales calculados\n\n');
261
262 % 8.7 Combinación SRSS
263
264 fprintf('Aplicando combinación SRSS...\n');
265
266 Umax_SRSS_X = sqrt(sum((phi * diag(Sd_X)).^2, 2));
267 Umax_SRSS_Y = sqrt(sum((phi * diag(Sd_Y)).^2, 2));
268 Umax_SRSS_Z = sqrt(sum((phi * diag(Sd_Z)).^2, 2));
269
270 fprintf('Combinación SRSS aplicada\n\n');
271
272 % 8.8 Reconstruir vectores en el espacio reducido (post-diafragma)
273 % ngdl aquí es ngdl_reducido (sobreescribo en mrigidez.m).
274 % Estos vectores tienen longitud ngdl_reducido, NO ngdl_original.
275 % La expansión al espacio original se realiza en la Sección 8.10
276 % mediante la transformación gdl_mapping.T.
277
278 % 8.8.1 Superposición algebraica (paso intermedio, no es resultado de diseño)
279 Umax_reducido_X = zeros(ngdl, 1);
280 Umax_reducido_Y = zeros(ngdl, 1);
281 Umax_reducido_Z = zeros(ngdl, 1);
282
283 Umax_reducido_X(gdl_libres) = Umax_modal_X;
284 Umax_reducido_Y(gdl_libres) = Umax_modal_Y;
285 Umax_reducido_Z(gdl_libres) = Umax_modal_Z;
286
287 % 8.8.2 Combinación SRSS (resultado espectral válido en espacio reducido)
288 Umax_SRSS_reducido_X = zeros(ngdl, 1);
289 Umax_SRSS_reducido_Y = zeros(ngdl, 1);
290 Umax_SRSS_reducido_Z = zeros(ngdl, 1);
291
292 Umax_SRSS_reducido_X(gdl_libres) = Umax_SRSS_X;
293 Umax_SRSS_reducido_Y(gdl_libres) = Umax_SRSS_Y;
294 Umax_SRSS_reducido_Z(gdl_libres) = Umax_SRSS_Z;
295
296 % 8.9 Desplazamientos por nivel (corregido para usar bien los diafragmas)
297
298 fprintf('Desplazamientos espectrales por nivel:\n');
299
300 fprintf(' Nivel Altura Ux (cm) Uy (cm) Uz (cm)\n');
301
302
303 % 8.10 Expandir al espacio original (ngdl_original = 6*nN)

```

```

304 % La transformación T: ngdl_original x ngdl_reducido expande los
305 % desplazamientos del espacio condensado al espacio completo.
306 if exist('gdl_mapping', 'var')
307     % 8.10.1 CON DIAFRAGMA: Reconstruir todos los GDL usando transformación
308     fprintf(' (Reconstruyendo desplazamientos con diafragma...)\n');
309     Umax_SRSS_completo_X = gdl_mapping.T * Umax_SRSS_reducido_X;
310     Umax_SRSS_completo_Y = gdl_mapping.T * Umax_SRSS_reducido_Y;
311     Umax_SRSS_completo_Z = gdl_mapping.T * Umax_SRSS_reducido_Z;
312 else
313     % 8.10.2 SIN DIAFRAGMA: El espacio reducido ya es el espacio original
314     Umax_SRSS_completo_X = Umax_SRSS_reducido_X;
315     Umax_SRSS_completo_Y = Umax_SRSS_reducido_Y;
316     Umax_SRSS_completo_Z = Umax_SRSS_reducido_Z;
317 end
318
319 % 8.11 Reorganizar desplazamientos (ahora con tamaño correcto: ngdl_original /
320     1)
321 U_nodes_X = reshape(Umax_SRSS_completo_X, 6, nN)';
322 U_nodes_Y = reshape(Umax_SRSS_completo_Y, 6, nN)';
323 U_nodes_Z = reshape(Umax_SRSS_completo_Z, 6, nN)';
324
325 % 8.11.1 Tabla de desplazamientos espectrales SRSS por nivel
326 fprintf('=====\n');
327 fprintf(' Nivel Altura Ux (cm) Uy (cm)\n');
328 for iNiv = 0:nNiv
329     z_nivel = iNiv * h;
330     nodos_nivel = [];
331     for iP = 1:(nProf+1)
332         for iC = 1:(nCruj+1)
333             idx = nodeIndex(iNiv+1, iC, iP, nNiv, nCruj);
334             nodos_nivel = [nodos_nivel, idx];
335         end
336     end
337     Ux_niv = mean(U_nodes_Y(nodos_nivel, 1)) * 100;
338     Uy_niv = mean(U_nodes_Y(nodos_nivel, 2)) * 100;
339     fprintf(' %2d %6.2f %8.4f %8.4f\n', iNiv, z_nivel, Ux_niv, Uy_niv);
340 end
341 fprintf('=====\n\n');
342
343 % 8.12 Gráfica del espectro
344 fprintf('Generando gráfica del espectro...\n');
345
346 figure('Name', 'Espectro de Diseño SASID', ...
347     'NumberTitle', 'off', ...
348     'Position', [100, 100, 900, 600]);
349
350 % 8.13 Espectro del SASID
351 plot(T_sasid, Sa_sasid, 'b-', 'LineWidth', 2.5);
352 hold on; grid on;
353
354 % 8.13.1 Si hay reducción adicional, mostrarla
355 if Q_adicional > 1.0
356     plot(T_sasid, Sa_sasid/Q_adicional, 'r--', 'LineWidth', 2);
357 end

```

```

358
359 % 8.13.2 Marcar períodos modales
360 for i = 1:min(6, n_modos)
361     plot(periodos(i), Sa(i)/9.81, 'ko', ...
362         'MarkerSize', 12, 'MarkerFaceColor', 'k');
363     text(periodos(i), Sa(i)/9.81, sprintf(' M%d', i), ...
364         'FontSize', 9, 'VerticalAlignment', 'bottom');
365 end
366
367 xlabel('Período T [s]', 'FontSize', 12, 'FontWeight', 'bold');
368 ylabel('Aceleración espectral Sa [g]', 'FontSize', 12, 'FontWeight', 'bold');
369
370 if Q_adicional > 1.0
371     title(sprintf('Espectro SASID (Q base=%.1f, Q diseño=%.1f)', Q_sasid,
372         Q_efectivo), ...
373         'FontSize', 14, 'FontWeight', 'bold');
374     legend({sprintf('SASID (Q=%.1f)', Q_sasid), ...
375         sprintf('Diseño (Q=%.1f)', Q_efectivo), ...
376         'Modos'}, 'Location', 'best', 'FontSize', 11);
377 else
378     title(sprintf('Espectro SASID (Q=%.1f)', Q_sasid), ...
379         'FontSize', 14, 'FontWeight', 'bold');
380     legend({sprintf('SASID (Q=%.1f)', Q_sasid), 'Modos'}, ...
381         'Location', 'best', 'FontSize', 11);
382 end
383
384 xlim([0, max(5, max(periodos(1:min(6,n_modos)))+0.5)]);
385 ylim([0, max(Sa_sasid)*1.1]);
386
387 hold off;
388 fprintf('Gráfica generada\n\n');
389
390 %% 9. Visualización de deformada espectral
391
392 fprintf('Generando visualización de deformada...\n');
393
394 % 9.1 Calcular factor de amplificación
395 % 9.1.1 Altura total del edificio
396 altura_total = nNiv * h;
397
398 % 9.1.2 Deformación máxima (usar SRSS en espacio reducido)
399 deformacion_maxima = max([
400     max(abs(Umax_SRSS_reducido_X)), ...
401     max(abs(Umax_SRSS_reducido_Y)), ...
402     max(abs(Umax_SRSS_reducido_Z))
403 ]);
404
405 % 9.1.3 Factor de amplificación
406 factor_amp = 10;
407
408 fprintf(' Factor de amplificación: %.1f--\n', factor_amp);
409 fprintf(' Deformación máxima: %.3f m\n\n', deformacion_maxima);
410
411 % 9.2 Umax_SRSS_completo_X/Y/Z ya fue calculado en la Sección 8.10.

```

```

412 % No se recalcula aquí para evitar inconsistencias si se modifica el código.
413
414 % 9.3 Reorganizar desplazamientos por nodo
415 % Cada U_nodos_SRSS_* tiene dimensión nNx6: filas=nodos, columnas=GDL (Ux,Uy,Uz,
    Rx,Ry,Rz)
416 U_nodos_SRSS_X = reshape(Umax_SRSS_completo_X, 6, nN)';
417 U_nodos_SRSS_Y = reshape(Umax_SRSS_completo_Y, 6, nN)';
418 U_nodos_SRSS_Z = reshape(Umax_SRSS_completo_Z, 6, nN)';
419
420 % 9.4 Nodos deformados (escala 1:1)
421 % CRITERIO DE DEFORMADA: La deformada se visualiza para la excitación en Y,
422 % que es la dirección dominante para estructuras regulares en planta.
423 % Para cada nodo se extraen las tres componentes traslacionales del mismo
424 % vector de respuesta (excitación Y), garantizando coherencia física.
425 % Cambiar U_nodos_SRSS_Y por U_nodos_SRSS_X o U_nodos_SRSS_Z según la
426 % dirección de análisis de interés.
427 coords_def = zeros(nN, 3);
428 for i = 1:nN
429     coords_def(i, 1) = N(i).x + U_nodos_SRSS_Y(i, 1); % Ux de respuesta Y
430     coords_def(i, 2) = N(i).y + U_nodos_SRSS_Y(i, 2); % Uy de respuesta Y
431     coords_def(i, 3) = N(i).z + U_nodos_SRSS_Y(i, 3); % Uz de respuesta Y
432 end
433
434 % 9.5 Nodos deformados (amplificados)
435 coords_def_amp = zeros(nN, 3);
436 for i = 1:nN
437     coords_def_amp(i, 1) = N(i).x + factor_amp * U_nodos_SRSS_Y(i, 1);
438     coords_def_amp(i, 2) = N(i).y + factor_amp * U_nodos_SRSS_Y(i, 2);
439     coords_def_amp(i, 3) = N(i).z + factor_amp * U_nodos_SRSS_Y(i, 3);
440 end
441
442 % 9.6 Crear figura interactiva
443
444 fig = figure('Name', 'Deformada Espectral', ...
445             'NumberTitle', 'off', ...
446             'Position', [50, 50, 1200, 800]);
447
448 ax = axes('Parent', fig);
449 hold(ax, 'on');
450 grid(ax, 'on');
451 axis(ax, 'equal');
452 view(ax, 40, 20);
453 xlabel(ax, 'X [m]', 'FontSize', 12);
454 ylabel(ax, 'Y [m]', 'FontSize', 12);
455 zlabel(ax, 'Z [m]', 'FontSize', 12);
456 title(ax, sprintf('Deformada Espectral (Factor: %.0f-)', factor_amp), ...
457       'FontSize', 14, 'FontWeight', 'bold');
458
459 % 9.7 Dibujar inicialmente con amplificación
460 plot_deformada_espectral(true, ax, N, coords_def, coords_def_amp, Barras,
    factor_amp);
461
462 % 9.8 Botón interactivo
463 uicontrol('Parent', fig, ...
464         'Style', 'togglebutton', ...

```

```

465     'String', 'Amplificación ON/OFF', ...
466     'Position', [20, 20, 180, 40], ...
467     'Value', 1, ...
468     'FontSize', 11, ...
469     'Callback', @(src, ~) plot_deformada_espectral( ...
470         get(src, 'Value') == 1, ax, N, coords_def, coords_def_amp, Barras,
            factor_amp));
471
472 fprintf('Visualización interactiva generada\n\n');
473
474 function plot_deformada_espectral(ampli, ax, N, coords_def, coords_def_amp,
    Barras, factor_amp)
475     % 9.8.1 Limpiar ejes
476     cla(ax);
477     hold(ax, 'on');
478
479     % 9.8.2 Colores
480     colorOriginal = [0.3, 0.3, 0.3]; % Gris oscuro
481     colorDeformada = [1.0, 0.2, 0.2]; % Rojo
482
483     % 9.8.3 Dibujar estructura original
484
485     for e = 1:length(Barras)
486         ni = Barras(e).nodol;
487         nj = Barras(e).nodo2;
488
489         plot3(ax, [N(ni).x, N(nj).x], ...
490             [N(ni).y, N(nj).y], ...
491             [N(ni).z, N(nj).z], ...
492             '-', 'Color', colorOriginal, 'LineWidth', 1.5);
493     end
494
495     % 9.8.4 Dibujar estructura deformada
496     if ampli
497         coords_plot = coords_def_amp; % Amplificada
498         titulo_extra = sprintf(' (Amplificación %.0f)', factor_amp);
499     else
500         coords_plot = coords_def; % Escala real
501         titulo_extra = ' (Escala 1:1)';
502     end
503
504     for e = 1:length(Barras)
505         ni = Barras(e).nodol;
506         nj = Barras(e).nodo2;
507
508         plot3(ax, [coords_plot(ni, 1), coords_plot(nj, 1)], ...
509             [coords_plot(ni, 2), coords_plot(nj, 2)], ...
510             [coords_plot(ni, 3), coords_plot(nj, 3)], ...
511             '--', 'Color', colorDeformada, 'LineWidth', 2.5);
512     end
513
514     % 9.8.5 Actualizar título
515     title(ax, ['Deformada Espectral', titulo_extra], ...
516         'FontSize', 14, 'FontWeight', 'bold');
517

```

```

518 % 9.8.6 Leyenda
519 legend(ax, {'Original', 'Deformada'}, 'Location', 'best', 'FontSize', 11);
520
521 hold(ax, 'off');
522 end
523
524 %% 10. Parámetros para Análisis Tiempo-Historia
525 fprintf('Configurando análisis tiempo-historia...\n');
526
527 % 10.1 Amortiguamiento modal (5% crítico)
528 xi = 0.05;
529
530 fprintf(' Amortiguamiento: %.1f%% crítico\n', xi * 100);
531 fprintf(' Modos a usar: %d\n\n', n_modos);
532
533 %% 11. Cargar acelerograma sísmico
534 fprintf('Cargando acelerograma...\n');
535
536 % 11.1 Tipo de acelerograma
537 opcion_ag = 'real'; % Opciones: 'sintetico' o 'real'
538
539 switch opcion_ag
540 case 'sintetico'
541 % 11.1.1 Acelerograma sintético (para pruebas)
542 dt = 0.01;
543 t_total = 20;
544 t = 0:dt:t_total;
545 ag = -0.4 * sin(2*pi*1.5*t) .* exp(-0.2*t);
546
547 fprintf(' Tipo: Acelerograma sintético\n');
548 fprintf(' Duración: %.1f s\n', t_total);
549
550 case 'real'
551
552 % 11.2 Acelerograma Real - 19 SEPTIEMBRE 2017 (SCT2)
553 % Cargar archivo
554 data = load('19S_SCT2.txt');
555 ag = data(:) * 0.01; % Convertir Gal a m/s^2
556 dt = 0.01; % 100 Hz
557 t = (0:length(ag)-1) * dt;
558
559 fprintf(' Tipo: Acelerograma real (SCT2 - 19/09/2017)\n');
560 fprintf(' (Convertido de Gal a m/s^2)\n');
561 fprintf(' Duración: %.1f s\n', t(end));
562 fprintf(' Paso de tiempo: %.4f s\n', dt);
563 fprintf(' Puntos: %d\n', length(ag));
564 fprintf(' Aceleración máxima: %.3f m/s^2 (%.2f g)\n', ...
565         max(abs(ag)), max(abs(ag))/9.81);
566
567 otherwise
568 error('Opción inválida. Usa ''sintetico'' o ''real''.');
569 end
570
571 nT = length(t);
572

```

```

573 fprintf('\n');
574
575 % 11.3 Visualizar acelerograma
576
577 fprintf('Generando gráfica del acelerograma...\n');
578
579 figure('Name', 'Acelerograma Sísmico', ...
580       'NumberTitle', 'off', ...
581       'Position', [100, 100, 1000, 400]);
582
583 plot(t, ag, 'b-', 'LineWidth', 1.2);
584 grid on;
585 xlabel('Tiempo [s]', 'FontSize', 12);
586 ylabel('Aceleración [m/s^2]', 'FontSize', 12);
587 title('Acelerograma 19 Septiembre 2017 SCT2', 'FontSize', 14, 'FontWeight', '
    bold');
588
589 % 11.4 Líneas de referencia
590 hold on;
591 plot([t(1), t(end)], [0, 0], 'k--', 'LineWidth', 0.5);
592 yline(max(abs(ag)), 'r--', sprintf('Max: %.2f m/s^2', max(abs(ag))));
593 yline(-max(abs(ag)), 'r--');
594 hold off;
595
596 fprintf('Gráfica generada\n\n');
597
598
599 %% 12. Integración temporal - Método de Newmark-beta
600 fprintf('Iniciando integración temporal (Newmark)...\n');
601
602 % 12.1 Inicializar matrices de respuesta modal
603 q = zeros(n_modos, nT);
604 qd = zeros(n_modos, nT);
605 qdd = zeros(n_modos, nT);
606
607 gamma_newmark = 0.5;
608 beta_newmark = 0.25;
609
610 fprintf(' Parámetros Newmark: g=%.2f, b^2=%.2f\n', gamma_newmark, beta_newmark);
611
612 % 12.2 Seleccionar dirección de excitación
613 direccion_excitacion = 'Y'; % CAMBIAR AQUÍ: 'X', 'Y', o 'Z'
614
615 switch direccion_excitacion
616     case 'X'
617         Gamma_excitacion = Gamma_X;
618         fprintf(' Dirección de excitación: X\n');
619     case 'Y'
620         Gamma_excitacion = Gamma_Y;
621         fprintf(' Dirección de excitación: Y\n');
622     case 'Z'
623         Gamma_excitacion = Gamma_Z;
624         fprintf(' Dirección de excitación: Z\n');
625     otherwise
626         error('Dirección inválida');

```

```

627 end
628
629 fprintf(' Integrando %d modos...\n', n_modos);
630
631
632 % 12.3 Integración por modo
633 for i = 1:n_modos
634     omega_i = omega(i);
635     m_modal = 1;
636     k_modal = omega_i^2;
637     c_modal = 2 * xi * omega_i * m_modal;
638
639     Gamma_i = Gamma_excitacion(i); % USA LA DIRECCIÓN SELECCIONADA
640
641     a0 = m_modal / (beta_newmark * dt^2);
642     a1 = c_modal * gamma_newmark / (beta_newmark * dt);
643     a2 = m_modal / (beta_newmark * dt);
644     a3 = c_modal * (gamma_newmark / beta_newmark - 1);
645     a4 = m_modal * (1 / (2 * beta_newmark) - 1);
646     a5 = c_modal * dt * (gamma_newmark / (2 * beta_newmark) - 1);
647
648     k_eff = k_modal + a0 + a1;
649
650     for j = 1:(nT - 1)
651         p_modal = -Gamma_i * ag(j+1);
652
653         p_eff = p_modal + a0 * q(i, j) + a2 * qd(i, j) + a4 * qdd(i, j) + ...
654             a1 * q(i, j) + a3 * qd(i, j) + a5 * qdd(i, j);
655
656         q(i, j+1) = p_eff / k_eff;
657
658         qd(i, j+1) = gamma_newmark / (beta_newmark * dt) * (q(i, j+1) - q(i, j)) +
659             ...
660             (1 - gamma_newmark / beta_newmark) * qd(i, j) + ...
661             dt * (1 - gamma_newmark / (2 * beta_newmark)) * qdd(i, j);
662
663         qdd(i, j+1) = 1 / (beta_newmark * dt^2) * (q(i, j+1) - q(i, j)) - ...
664             1 / (beta_newmark * dt) * qd(i, j) - ...
665             (1 / (2 * beta_newmark) - 1) * qdd(i, j);
666
667     end
668
669     if mod(i, 3) == 0 || i == n_modos
670         fprintf(' Modo %d/%d completado\n', i, n_modos);
671     end
672 end
673
674 fprintf('Integración completada\n\n');
675
676 %% 13. Reconstrucción de desplazamientos físicos
677 fprintf('Reconstruyendo desplazamientos físicos...\n');
678
679 % 13.1 Inicializar vector global (incluye GDL restringidos = 0)
680 U_hist_global = zeros(ngdl, nT);

```

```

681
682 % 13.2 Transformar coordenadas modales a físicas
683 %  $u(t) = \Phi * q(t)$ 
684 U_hist_libres = phi * q; % Solo GDL libres
685
686 % 13.3 Asignar a vector global
687 U_hist_global(gdl_libres, :) = U_hist_libres;
688
689 fprintf('Desplazamientos reconstruidos\n');
690 fprintf(' Tamaño: %d GDL - %d pasos de tiempo\n\n', ngdl, nT);
691
692
693 % 13.4 Expandir al espacio original mediante transformación de diafragma.
694 % U_hist_completo: ngdl_original x nT = 6*nN x nT
695 % Esta es la única expansión de U_hist_completo en el script.
696 % Todas las secciones posteriores (estadísticas, gráficas, animación)
697 % usan esta misma variable sin recalcularla.
698 if exist('gdl_mapping', 'var')
699     % CON DIAFRAGMA: expandir espacio reducido -> espacio original
700     U_hist_completo = gdl_mapping.T * U_hist_global;
701     fprintf(' Tamaño expandido: %d GDL x %d pasos de tiempo\n\n', ...
702           size(U_hist_completo, 1), size(U_hist_completo, 2));
703 else
704     % SIN DIAFRAGMA: el espacio reducido ya es el espacio original
705     U_hist_completo = U_hist_global;
706 end
707
708 % 13.5 Desplazamiento máximo en cada dirección
709 U_nodes_hist = reshape(U_hist_completo, 6, nN, nT);
710
711 Ux_max = max(max(abs(squeeze(U_nodes_hist(1, :, :)))));
712 Uy_max = max(max(abs(squeeze(U_nodes_hist(2, :, :)))));
713 Uz_max = max(max(abs(squeeze(U_nodes_hist(3, :, :)))));
714
715 fprintf('Desplazamientos máximos:\n');
716 fprintf(' Ux: %.3f m = %.1f cm\n', Ux_max, Ux_max * 100);
717 fprintf(' Uy: %.3f m = %.1f cm\n', Uy_max, Uy_max * 100);
718 fprintf(' Uz: %.3f m = %.1f cm\n\n', Uz_max, Uz_max * 100);
719
720 % 13.6 Tabla de desplazamientos máximos tiempo-historia por nivel
721 fprintf('Desplazamientos máximos tiempo-historia por nivel:\n');
722 fprintf('===== \n');
723 fprintf(' Nivel Altura Ux_max(cm) Uy_max(cm) \n');
724 for iNiv = 0:nNiv
725     z_nivel = iNiv * h;
726     nodos_nivel = [];
727     for iP = 1:(nProf+1)
728         for iC = 1:(nCruj+1)
729             idx = nodeIndex(iNiv+1, iC, iP, nNiv, nCruj);
730             nodos_nivel = [nodos_nivel, idx];
731         end
732     end
733     Ux_max_niv = max(max(abs(squeeze(U_nodes_hist(1, nodos_nivel, :)))) * 100;
734     Uy_max_niv = max(max(abs(squeeze(U_nodes_hist(2, nodos_nivel, :)))) * 100;
735     fprintf(' %2d %6.2f %8.4f %8.4f\n', iNiv, z_nivel, Ux_max_niv, Uy_max_niv);

```

```

736 end
737 fprintf('=====\n\n');
738
739
740 %% 14. Visualización de Historia de Desplazamientos
741 fprintf('Generando gráficas de respuesta temporal...\n');
742
743 % 14.1 Selecciona nodo representativo
744 nodo_azotea = (nNiv + 1) * (nCruj + 1) * (nProf + 1); % Último nodo
745 % como en el análisis estático
746
747 fprintf(' Nodo seleccionado: %d (azotea)\n', nodo_azotea);
748
749 % 14.2 Gráficas de desplazamiento en múltiples direcciones
750 % 14.2.1 GDL del nodo de azotea
751 gdl_ux = 6 * (nodo_azotea - 1) + 1; % Ux
752 gdl_uy = 6 * (nodo_azotea - 1) + 2; % Uy
753 gdl_uz = 6 * (nodo_azotea - 1) + 3; % Uz
754
755 % 14.3 Verificar que estén libres y obtener índices locales
756 idx_ux = find(gdl_libres == gdl_ux);
757 idx_uy = find(gdl_libres == gdl_uy);
758 idx_uz = find(gdl_libres == gdl_uz);
759
760 % 14.4 Crear figura con 4 subplots (X, Y, Z + acelerograma)
761 figure('Name', 'Historia de Desplazamiento - Todas las direcciones', ...
762     'NumberTitle', 'off', ...
763     'Position', [50, 50, 1400, 900]);
764
765
766 % 14.4.1 SUBPLOT 1: Desplazamiento en X
767
768 subplot(4, 1, 1);
769
770 if gdl_ux <= size(U_hist_completo, 1)
771     plot(t, U_hist_completo(gdl_ux, :) * 100, 'b-', 'LineWidth', 1.5);
772
773     % Marcar máximo
774     [umax_x, idx_max_x] = max(abs(U_hist_completo(gdl_ux, :)));
775     hold on;
776     plot(t(idx_max_x), U_hist_completo(gdl_ux, idx_max_x) * 100, 'ro', ...
777         'MarkerSize', 10, 'MarkerFaceColor', 'r');
778     text(t(idx_max_x), U_hist_completo(gdl_ux, idx_max_x) * 100, ...
779         sprintf(' Max: %.2f cm @ t=%.1fs', umax_x * 100, t(idx_max_x)), ...
780         'FontSize', 9, 'BackgroundColor', 'white');
781     hold off;
782 end
783
784 ylabel('U_x [cm]', 'FontSize', 11, 'FontWeight', 'bold');
785 title('Desplazamiento en X', 'FontSize', 11);
786 grid on;
787
788 % 14.4.2 SUBPLOT 2: Desplazamiento en Y
789
790 subplot(4, 1, 2);

```

```

791 if gdl_uy <= size(U_hist_completo, 1)
792     plot(t, U_hist_completo(gdl_uy, :) * 100, 'g-', 'LineWidth', 1.5);
793
794     [umax_y, idx_max_y] = max(abs(U_hist_completo(gdl_uy, :)));
795     hold on;
796     plot(t(idx_max_y), U_hist_completo(gdl_uy, idx_max_y) * 100, 'ro', ...
797         'MarkerSize', 10, 'MarkerFaceColor', 'r');
798     text(t(idx_max_y), U_hist_completo(gdl_uy, idx_max_y) * 100, ...
799         sprintf(' Max: %.2f cm @ t=%.1fs', umax_y * 100, t(idx_max_y)), ...
800         'FontSize', 9, 'BackgroundColor', 'white');
801     hold off;
802 end
803
804 ylabel('U_y [cm]', 'FontSize', 11, 'FontWeight', 'bold');
805 title('Desplazamiento en Y', 'FontSize', 11);
806 grid on;
807
808
809
810 % 14.4.3 SUBPLOT 3: Desplazamiento en Z
811
812 subplot(4, 1, 3);
813
814 if gdl_uz <= size(U_hist_completo, 1)
815     plot(t, U_hist_completo(gdl_uz, :) * 100, 'm-', 'LineWidth', 1.5);
816
817     [umax_z, idx_max_z] = max(abs(U_hist_completo(gdl_uz, :)));
818     hold on;
819     plot(t(idx_max_z), U_hist_completo(gdl_uz, idx_max_z) * 100, 'ro', ...
820         'MarkerSize', 10, 'MarkerFaceColor', 'r');
821     text(t(idx_max_z), U_hist_completo(gdl_uz, idx_max_z) * 100, ...
822         sprintf(' Max: %.2f cm @ t=%.1fs', umax_z * 100, t(idx_max_z)), ...
823         'FontSize', 9, 'BackgroundColor', 'white');
824     hold off;
825 end
826
827 ylabel('U_z [cm]', 'FontSize', 11, 'FontWeight', 'bold');
828 title('Desplazamiento en Z', 'FontSize', 11);
829 grid on;
830
831 % 14.4.4 SUBPLOT 4: Acelerograma de entrada
832
833 subplot(4, 1, 4);
834 plot(t, ag, 'r-', 'LineWidth', 1.0);
835 grid on;
836 xlabel('Tiempo [s]', 'FontSize', 12, 'FontWeight', 'bold');
837 ylabel('ag [m/s^2]', 'FontSize', 11, 'FontWeight', 'bold');
838 title(sprintf('Acelerograma 19S-2017 SCT2 (Dirección: %s)', direccion_excitacion
839     ), ...
840     'FontSize', 12, 'FontWeight', 'bold');
841
842 % 14.5 Marcar máximo del acelerograma
843 [ag_max, idx_ag_max] = max(abs(ag));
844 hold on;

```

```

844 plot(t(idx_ag_max), ag(idx_ag_max), 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'r
      ');
845 hold off;
846
847 fprintf('Gráficas generadas (X, Y, Z)\n\n');
848
849
850 % 14.6 Resumen de desplazamientos máximos
851 fprintf('Desplazamientos máximos en nodo %d:\n', nodo_azotea);
852 if ~isempty(idx_ux)
853     fprintf(' Ux: %.2f cm @ t=%.1f s\n', umax_x * 100, t(idx_max_x));
854 end
855 if ~isempty(idx_uy)
856     fprintf(' Uy: %.2f cm @ t=%.1f s\n', umax_y * 100, t(idx_max_y));
857 end
858 if ~isempty(idx_uz)
859     fprintf(' Uz: %.2f cm @ t=%.1f s\n\n', umax_z * 100, t(idx_max_z));
860 end
861
862
863
864 %% 15. Animación 3D de respuesta dinámica
865 fprintf('Preparando animación 3D...\n');
866
867
868 % 15.1 Configuración de animación
869 factor_anim = 5e3; % Factor alto
870 salto_frames = 10; % Saltar 10 frames
871 duracion_max = 60; % Solo primeros 60 segundos
872
873
874 % 15.2 Recortar historia a primeros 60s
875 idx_max = find(t <= duracion_max, 1, 'last');
876 if isempty(idx_max)
877     idx_max = length(t);
878 end
879
880 fprintf(' Duración: %.1f s\n', t(idx_max));
881 fprintf(' Factor amplificación: %.0e\n', factor_anim);
882 fprintf(' Salto de frames: %d\n', salto_frames);
883 fprintf(' Frames totales: %d\n\n', ceil(idx_max/salto_frames));
884
885
886 % 15.3 Crear figura y configurar ejes
887 fig_anim = figure('Name', 'Animación Historia de Tiempo 3D', ...
888                 'NumberTitle', 'off', ...
889                 'Position', [50, 50, 1200, 800]);
890
891 ax = axes('Parent', fig_anim);
892 hold(ax, 'on');
893 axis(ax, 'equal');
894 grid(ax, 'on');
895 view(ax, 40, 20);
896 xlabel(ax, 'X [m]', 'FontSize', 12);
897 ylabel(ax, 'Y [m]', 'FontSize', 12);

```

```

898 xlabel(ax, 'Z [m]', 'FontSize', 12);
899
900 % 15.4 Límites fijos (calculados una vez)
901 coords_x = [N.x];
902 coords_y = [N.y];
903 coords_z = [N.z];
904
905 margen_x = 0.2 * (max(coords_x) - min(coords_x));
906 margen_y = 0.2 * (max(coords_y) - min(coords_y));
907 margen_z = 0.25 * (max(coords_z) - min(coords_z));
908
909 xlim(ax, [min(coords_x) - margen_x, max(coords_x) + margen_x]);
910 ylim(ax, [min(coords_y) - margen_y, max(coords_y) + margen_y]);
911 zlim(ax, [min(coords_z) - margen_z, max(coords_z) + margen_z]);
912
913
914 % 15.5 Estructura para nodos animados
915 N_anim = N; % Copiar estructura original
916
917
918 % 15.6 Loop de animación - Tiempo Real del acelerograma
919 fprintf('Ejecutando animación...\n');
920 fprintf('(Presiona Ctrl+C para detener)\n\n');
921
922 % 15.7 Configuración de tiempo real
923 velocidad = 1.0;
924
925 % Opciones:
926 % velocidad = 1.0; % Tiempo real (1 segundo simulado = 1 segundo real)
927 % velocidad = 2.0; % Doble velocidad (60s en 30s)
928 % velocidad = 5.0; % Muy rápido (60s en 12s)
929 % velocidad = 0.5; % Cámara lenta (60s en 120s)
930
931
932 fprintf(' Velocidad: %.1f-- (1 s simulado = %.2f s reales)\n', velocidad, 1/
    velocidad);
933 fprintf(' Tiempo simulado por frame: %.2f s\n', salto_frames * dt);
934 fprintf(' Pausa objetivo entre frames: %.3f s\n\n', (salto_frames * dt) /
    velocidad);
935
936 % 15.8 Iniciar cronómetro
937 tic_inicio = tic;
938 tic_frame = tic;
939
940 j = 1;
941 frame_count = 0;
942
943 while j <= idx_max
944     % 15.8.1 Verificar que ventana sigue abierta
945     if ~isvalid(fig_anim)
946         fprintf('Ventana cerrada\n');
947         break;
948     end
949
950     % 15.8.2 Actualizar gráfica

```

```

951     cla(ax);
952
953 % 15.9 Actualizar posiciones de nodos
954     for i = 1:nN
955         dofx = 6 * (i - 1) + 1;
956         dofy = dofx + 1;
957         dofz = dofx + 2;
958
959 % 15.9.1 Usar U_hist_completo (ya reconstruido previamente)
960         N_anim(i).x = N(i).x + factor_anim * U_hist_completo(dofx, j);
961         N_anim(i).y = N(i).y + factor_anim * U_hist_completo(dofy, j);
962         N_anim(i).z = N(i).z + factor_anim * U_hist_completo(dofz, j);
963     end
964
965 % 15.9.2 Dibujar elementos
966     for e = 1:length(Barras)
967         ni = Barras(e).nodo1;
968         nj = Barras(e).nodo2;
969
970         plot3(ax, [N_anim(ni).x, N_anim(nj).x], ...
971             [N_anim(ni).y, N_anim(nj).y], ...
972             [N_anim(ni).z, N_anim(nj).z], ...
973             'b-', 'LineWidth', 2);
974     end
975
976 % 15.9.3 Actualizar título con tiempo transcurrido
977     tiempo_transcurrido = toc(tic_inicio);
978     title(ax, sprintf('t = %.2f s | Transcurrido: %.1f s', t(j),
979         tiempo_transcurrido), ...
980         'FontSize', 14, 'FontWeight', 'bold');
981
982 % 15.10 Sincronización con tiempo real
983     drawnow; % Forzar actualización de gráfica
984
985 % 15.10.1 Calcular cuánto tiempo DEBERÍA haber pasado
986     frame_count = frame_count + 1;
987     tiempo_simulado = t(j);
988     tiempo_objetivo = tiempo_simulado / velocidad; % Tiempo real que debería
989         llevar
990
991 % 15.11 Tiempo real transcurrido
992     tiempo_real = toc(tic_inicio);
993
994 % 15.11.1 Si está adelantado, pausar
995     if tiempo_real < tiempo_objetivo
996         pause(tiempo_objetivo - tiempo_real);
997     end
998
999 % 15.11.2 Avanzar al siguiente frame
1000     j = j + salto_frames;
1001 end
1002 fprintf('\n Animación completada\n');
1003 fprintf(' Tiempo simulado: %.1f s\n', t(min(j-1, idx_max)));

```

```

1004 fprintf(' Tiempo real transcurrido: %.1f s\n', toc(tic_inicio));
1005 fprintf(' Frames renderizados: %d\n\n', frame_count);
1006
1007
1008 % NOTA: U_hist_completo fue calculado correctamente en la Sección 13
1009 % (antes de la animación). El bloque que existía aquí era redundante
1010 % y posterior al uso de la variable, por lo que se elimina.
1011
1012 %% Cálculo de carga distribuida equivalente para validación en SAP2000
1013 % La masa de losa se convierte a carga lineal uniforme sobre vigas
1014 % para replicar la distribución nodal de MATLAB en SAP2000 (sin shells).
1015 m_losa_por_m2 = 600; % [kg/m^2] densidad superficial de losa
1016 A_planta = (nCruj*w) * (nProf*L); % [m^2] área de planta
1017 m_losa_total = m_losa_por_m2 * A_planta * nNiv; % [kg] masa total losas
1018 n_pisos = nNiv;
1019 L_vigas_piso = nProf*(nCruj+1)*w + nCruj*(nProf+1)*L; % [m] longitud total de
    vigas/piso
1020 w_losa_SAP = (m_losa_total / n_pisos) * 9.81 / L_vigas_piso / 1000; % [kN/m]
1021 % -----
1022 fprintf('\n CARGA EQUIVALENTE DE LOSA PARA SAP2000\n');
1023 fprintf(' -----\n');
1024 fprintf(' Masa losa total: %.0f kg (%.1f ton)\n', m_losa_total, m_losa_total
    /1000);
1025 fprintf(' Masa losa por piso: %.0f kg (%.1f ton)\n', m_losa_total/n_pisos,
    m_losa_total/n_pisos/1000);
1026 fprintf(' Long. vigas por piso: %.1f m\n', L_vigas_piso);
1027 fprintf(' Carga distribuida: %.2f kN/m\n', w_losa_SAP);
1028 fprintf(' -> Usar en SAP2000: Assign > Frame Loads > Distributed\n');
1029 fprintf(' Load Pattern: LOSA, Direction: Gravity, Uniform: %.2f kN/m\n',
    w_losa_SAP);
1030 fprintf(' -> Activar: Element Self Mass + Specified Load Pattern LOSA x1.0\n\n')
    ;
1031
1032 %% Funciones locales
1033 % Función para calcular el índice del nodo en la malla 3D
1034 % iN: nivel (1 a nNiv+1)
1035 % iC: crujía en Y (1 a nCruj+1)
1036 % iP: profundidad en X (1 a nProf+1)
1037 % nNiv: número total de niveles
1038 % nCruj: número total de crujías en Y
1039
1040 function idx = nodeIndex(iN, iC, iP, nNiv, nCruj)
1041
1042     offset_profundidad = (iP - 1) * (nCruj + 1) * (nNiv + 1);
1043     offset_crujia = (iC - 1) * (nNiv + 1);
1044     offset_nivel = iN;
1045
1046     idx = offset_profundidad + offset_crujia + offset_nivel;
1047 end

```

B.5. analisis_ise_superposicion.m

```

1  % ANÁLISIS DE INTERACCIÓN SUELO-ESTRUCTURA
2  % Método simplificado: Modificación de periodos
3  %
4  % ESTRATEGIA: Trabajar con K_red y M_red (igual que analisis_dinamico.m)
5  % y modificar SOLO los términos diagonales correspondientes a la base
6  %
7
8  clear all; clc; close all;
9
10 fprintf('\n');
11 fprintf('ANÁLISIS ISE - Método de Superposición \n');
12 fprintf('\n');
13
14 %% 0. Cargar modelo (igual que analisis_dinamico.m)
15 fprintf('PASO 0: Cargando modelo estructural...\n');
16 fprintf('-----\n');
17
18 if ~exist('K_red', 'var')
19     fprintf(' Ejecutando scripts previos...\n');
20     run('marco3d_editable.m');
21     run('mrigidez.m');
22 end
23
24 fprintf('-----');
25 fprintf(' - Nodos: %d\n', nN);
26 fprintf(' - GDLs libres: %d\n', length(gdl_libres));
27 fprintf('\n');
28
29 % Compatibilidad con modelos sin diafragma rígido.
30 % Sin diafragma: gdl_libres contiene índices en espacio original 1..6*nN,
31 % por lo que gdl_mapping se define como identidad para que el resto del
32 % script funcione sin modificaciones.
33 % Con diafragma: gdl_mapping fue creado por mrigidez.m y no se sobrescribe.
34 if ~exist('gdl_mapping', 'var')
35     ngdl_orig = 6 * nN;
36     gdl_mapping.independientes = 1:ngdl_orig;
37     gdl_mapping.T = speye(ngdl_orig);
38     gdl_mapping.ngdl_original = ngdl_orig;
39     gdl_mapping.ngdl_reducido = ngdl;
40     fprintf(' Modo ISE: SIN diafragma (gdl_mapping = identidad)\n\n');
41 else
42     fprintf(' Modo ISE: CON diafragma (condensación master-slave activa)\n\n');
43 end
44
45
46 %% 1. Parámetros del suelo
47 fprintf('PASO 1: Parámetros Zona III\n');
48 fprintf('-----\n');
49
50 Vs_suelo = 80;
51 rho_suelo = 1250;
52 nu_suelo = 0.499;
53 beta_suelo = 0.03;

```

```

54 H_suelo = 50;
55 G_suelo = rho_suelo * Vs_suelo^2;
56
57 fprintf(' Vs=%.0f m/s, rho=%.0f kg/m^3, nu=%.3f, G=%.0f kPa\n', ...
58         Vs_suelo, rho_suelo, nu_suelo, G_suelo/1000);
59 fprintf(' Ref: NTC-Cimentaciones (2017)\n\n');
60
61 %% 2. Geometría cimentación
62 fprintf('PASO 2: Cimentación\n');
63 fprintf('-----\n');
64
65 L_cim = nCruj * L;
66 B_cim = nProf * w;
67 A_cim = L_cim * B_cim;
68 r0_cim = sqrt(A_cim/pi);
69
70 fprintf(' Losa: %.1f-%.1f m, r0=%.2f m\n', L_cim, B_cim, r0_cim);
71 fprintf('\n');
72
73 %% 3. Nodos de base
74 fprintf('PASO 3: Identificar nodos base\n');
75 fprintf('-----\n');
76
77 tol = 1e-6;
78 nodos_base = find(abs(xyz(:,3)) < tol);
79
80 fprintf(' %d nodos en z=0\n', length(nodos_base));
81 fprintf('\n');
82
83 %% 4. Análisis modal BASE RÍGIDA (igual que analisis_dinamico.m)
84 fprintf('PASO 4: Análisis modal BASE RÍGIDA\n');
85 fprintf('-----\n');
86
87 n_modos = min(12, length(gdl_libres));
88 [phi_br, lambda_br] = eigs(K_red, M_red, n_modos, 'smallestabs');
89
90 [lambda_sorted, idx] = sort(diag(lambda_br));
91 phi_br = phi_br(:, idx);
92 omega_br = sqrt(real(lambda_sorted)); % real() protege contra parte imaginaria
93     residual (error numérico)
94 T_br = 2*pi ./ omega_br;
95 f_br = 1 ./ T_br;
96
97 fprintf(' Primeros 3 modos:\n');
98 for i = 1:min(3,n_modos)
99     fprintf(' Modo %d: T=%.3f s, f=%.3f Hz\n', i, T_br(i), f_br(i));
100 end
101 fprintf('\n');
102
103 %% 5. Impedancias dinámicas del suelo - parámetros independientes de frecuencia
104 fprintf('PASO 5: Geometría de cimentación e impedancias estáticas\n');

```

```

104 fprintf('-----\n');
105 fprintf(' Ref: Gazetas (1991); Avilés López et al. (2016), Sec. 6.3.2\n');
106 fprintf(' Formulación: disco circular rígido sobre semiespacio elástico\n');
107 fprintf(' (Luco & Westmann 1971, extendida por Gazetas 1983, 1991)\n\n');
108 fprintf(' NOTA: Los coeficientes dinámicos (Ch, Cr) dependen de la\n');
109 fprintf(' frecuencia efectiva T~ y se calculan en el PASO 6 mediante\n');
110 fprintf(' iteración (NTC-Sismo 2020, Sec. 8.2.3).\n\n');
111
112 % 5.1 Radios equivalentes (independientes de frecuencia)
113 % Radio traslacional: círculo de igual área a la planta de cimentación
114 % Ref: NTC-Sismo 2020, ec. 8.2.13; Avilés López et al. (2016), Sec. A4.4
115 % r0 = sqrt(Aπ/)
116 % (ya calculado en PASO 2 como r0_cim)
117
118 % Radio rotacional: círculo de igual momento de inercia a la planta
119 % Ref: NTC-Sismo 2020, ec. 8.2.14; Avilés López et al. (2016), Sec. A4.4
120 % r0r = (Iπ/)^ (1/4) donde I = momento de inercia de la planta respecto
121 % al eje centroidal perpendicular a la dirección de análisis
122 Ix_cim = (L_cim * B_cim^3) / 12; % Inercia respecto a eje X (cabecero en Y)
123 Iy_cim = (B_cim * L_cim^3) / 12; % Inercia respecto a eje Y (cabecero en X)
124 r0r_x = (4*Ix_cim / pi)^(1/4); % NTC-Sismo 2020 ec. 8.2.14: r = (4Iπ/)^ (1/4)
125 r0r_y = (4*Iy_cim / pi)^(1/4);
126 r0r = mean([r0r_x, r0r_y]); % Promedio (edificio simétrico en planta)
127
128 fprintf(' Radios equivalentes:\n');
129 fprintf(' r0 (traslación): %.3f m [= sqrt(Aπ/)]\n', r0_cim);
130 fprintf(' r0r (rotación X): %.3f m [= (4Ixπ/)^ (1/4)]\n', r0r_x);
131 fprintf(' r0r (rotación Y): %.3f m [= (4Iyπ/)^ (1/4)]\n', r0r_y);
132 fprintf(' r0r (promedio): %.3f m\n\n', r0r);
133
134 % 5.2 Rigideces estáticas (independientes de frecuencia)
135 % Estas son las rigideces para excitación cuasi-estática ω( -> 0).
136 % Los coeficientes dinámicos kh y kr modifican estas rigideces para ω > 0,
137 % pero para cimentaciones superficiales en semiespacio kh ≈ kr ≈ 1
138 % (Gazetas 1991, Fig. 3), por lo que Kh_static ≈ Khω() para a0 < 0.5.
139 % Ref: Gazetas (1991), ecs. (3.13) y (3.15)
140
141 Kh_static = (8 * G_suelo * r0_cim) / (2 - nu_suelo); % [N/m]
142 Kr_static = (8 * G_suelo * r0r^3) / (3*(1 - nu_suelo)); % [N*m/rad]
143
144 fprintf(' Rigideces estáticas:\n');
145 fprintf(' Kh = 8·G·r0/(2-nu) = %.4e N/m\n', Kh_static);
146 fprintf(' Kr = 8·G·r0r^3/[3(1-nu)] = %.4e N*m/rad\n\n', Kr_static);
147
148 % 5.3 Coeficientes de amortiguamiento radiativo (función auxiliar)
149 % Ch y Cr se evalúan en la frecuencia omegal dentro del lazo iterativo.
150 % Se definen aquí como funciones anónimas para uso en PASO 6.
151 % Ch(omegal) = Kh · ch_coef · (omegal·r0/Vs) · (r0/Vs)
152 % Cr(omegal) = Kr · cr_coef · (omegal·r0r/Vs) · (r0r/Vs)
153 % Ref: Avilés López et al. (2016), ecs. 6.3.16-6.3.17; Gazetas (1991)
154 ch_coef = 0.6; % coeficiente traslación horizontal (Gazetas 1991, Tabla 3)
155 cr_coef = 0.4; % coeficiente cabecero (Gazetas 1991, Tabla 3)
156

```

```

157 Ch_fun = @(omega_eff) Kh_static * ch_coef * (omega_eff*r0_cim/Vs_suelo) * (
    r0_cim/Vs_suelo);
158 Cr_fun = @(omega_eff) Kr_static * cr_coef * (omega_eff*r0r/Vs_suelo) * (r0r/
    Vs_suelo);
159
160 fprintf(' Coeficientes de amortiguamiento (evaluados en iteración, ver PASO 6):\n'
    n');
161 fprintf(' ch = %.2f (traslación horizontal, Gazetas 1991)\n', ch_coef);
162 fprintf(' cr = %.2f (cabeceo, Gazetas 1991)\n\n', cr_coef);
163
164 % 5.4 Verificación: frecuencia resonante del estrato
165 % Para frecuencias menores que la fundamental del estrato, el
166 % amortiguamiento por radiación es prácticamente nulo.
167 % Ref: Avilés López et al. (2016), Sec. 6.3.2; NTC-Sismo 2020, Sec. 8.2
168 freq_resonante_estrato = Vs_suelo / (4 * H_suelo); % [Hz]
169 fprintf(' Verificación amortiguamiento por radiación:\n');
170 fprintf(' Frecuencia resonante del estrato: fs = %.4f Hz\n',
    freq_resonante_estrato);
171 fprintf(' Frecuencia fundamental estructura: fl = %.4f Hz\n', f_br(1));
172 if f_br(1) < freq_resonante_estrato
173     fprintf(' *** ADVERTENCIA: fl < fs -> radiación prácticamente nula.\n');
174     fprintf(' El amortiguamiento efectivo proviene del\n');
175     fprintf(' amortiguamiento histéretico del suelo (beta_s = %.2f).\n',
        beta_suelo);
176 else
177     fprintf(' OK: fl > fs -> amortiguamiento radiativo activo.\n');
178 end
179 fprintf('\n');
180
181 %% 6. Análisis modal BASE FLEXIBLE - Periodo y amortiguamiento efectivos
182 fprintf('PASO 6: Análisis modal BASE FLEXIBLE\n');
183 fprintf('-----\n'
    n');
184 fprintf(' MÉTODO: Periodo efectivo por suma de cuadrados (NTC-Sismo 2020)\n');
185 fprintf(' Ref principal: NTC-Sismo (2020), ecs. 8.2.7-8.2.12\n');
186 fprintf(' Ref secundaria: Avilés López et al. (2016), Sec. A4.3\n\n');
187
188 %-----
189 % FUNDAMENTO FÍSICO Y NORMATIVO
190 %
191 % El sistema suelo-estructura equivalente de un grado de libertad
192 % (Avilés & Pérez-Rocha, 1996; Wolf, 1985) tiene tres modos de
193 % deformación independientes:
194 % 1. Deformación de la superestructura: periodo Te (base rígida)
195 % 2. Traslación de cuerpo rígido de la cimentación: periodo Th
196 % 3. Rotación (cabeceo) de cuerpo rígido de la cimentación: periodo Tr
197 %
198 % El periodo efectivo del sistema acoplado se obtiene como (NTC ec. 8.2.7):
199 %  $T_{\sim}^2 = T_{\sim}^2 + T_1^2 + T_1^2$ 
200 %
201 % donde (NTC ecs. 8.2.8 y 8.2.9, D=0 para cimentación superficial):
202 %  $T_1 = 2\pi * \sqrt{W_{\sim} / (g * Kh)}$ 
203 %  $T_1 = 2\pi * \sqrt{W_{\sim} * H_{\sim}^2 / (g * Kr)}$ 
204 %
205 %  $W_{\sim} = \text{peso efectivo modal} = M_{\sim} * g$ 

```

```

206 % M~ = masa efectiva del modo fundamental (ec. 6.5.1 de Avilés López)
207 % H~ = altura efectiva del centro de masa equivalente
208 % D = enterramiento de la cimentación (= 0 para cimentación superficial)
209 %
210 % SUPUESTOS DECLARADOS:
211 % (a) Cimentación superficial: D = 0
212 % (b) Interacción cinemática despreciada: válido para cimentaciones
213 % superficiales con ondas de corte con incidencia vertical
214 % (Avilés & Pérez-Rocha, 2003; NTC-Sismo 2020, Sec. 8.1)
215 % (c) Modos superiores no modificados por ISE
216 % (NTC-Sismo 2020, Sec. 8.2; Avilés López et al. 2016, Sec. 6.5.3)
217 % (d) Impedancias evaluadas iterativamente en  $\omega_{al} = 2\pi/T$  hasta
218 % convergencia. Esto es el procedimiento riguroso de NTC-Sismo 2020,
219 % Sec. 8.2.3 y Avilés López et al. (2016), Sec. 6.5.2.1.
220 %-----
221
222 g = 9.81; % m/s^2
223 D_cim = 0; % [m] Enterramiento de la cimentación (superficial)
224
225 %-----
226 % DIRECCIÓN DE ANÁLISIS
227 % El acelerograma 19S-SCT2 se aplica en dirección Y (Norte-Sur).
228 % El modo dominante debe seleccionarse maximizando la masa efectiva en Uy.
229 % Cambiar a 'X' si se analiza en la otra dirección.
230 dir_analisis = 'Y'; % 'X' o 'Y'
231 %-----
232
233 % 6.1 Masa efectiva del modo fundamental
234 %  $M_{\sim} = \Sigma(\varphi_{m\cdot\sim})^2 / \Sigma(\varphi_{m\cdot\sim}^2)$ 
235 % donde  $\varphi_{\sim}$  es la amplitud del modo 1 en el nodo n (GDLs traslacionales).
236 % Ref: Wolf (1985); Avilés López et al. (2016), ec. 6.5.1
237 %
238 % Implementación: Se extraen los GDLs Ux del modo 1 normalizado por masa.
239 % La normalización por masa garantiza:  $\varphi^T M \varphi_1 = I \rightarrow$  se puede usar
    directamente.
240
241 % 6.1.1 Construir vector de influencia en X
242 % IMPORTANTE: gdl_libres contiene índices locales dentro del sistema
243 % reducido por diafragma (1..ngdl_reducido). Para identificar el TIPO
244 % de cada GDL (Ux, Uy, etc.) hay que recuperar el índice ORIGINAL en
245 % el espacio completo (1..ngdl_original = 6*nNodos) usando gdl_mapping.
246 %
247 % gdl_mapping.independientes(k) = índice original del GDL reducido k
248 % gdl_libres(i) = índice dentro del sistema reducido sin restricciones
249 %
250 % Proceso:
251 % índice reducido libre i
252 % -> índice en sistema con diafragma: gdl_mapping.independientes(gdl_libres(i))
253 % -> tipo GDL: mod(idx_original - 1, 6) + 1
254
255 n_dof_6 = size(M_red, 1);
256 J_x = zeros(n_dof_6, 1);
257 J_y = zeros(n_dof_6, 1);
258 idx_Ux = [];
259 idx_Uy = [];

```

```

260
261 % ESTRATEGIA: identificar tipo de GDL mediante coordenada z del nodo.
262 % Un GDL pertenece a un nodo libre (z > 0) o a la base (z = 0).
263 % Solo se activan GDLs traslacionales de nodos con z > 0.
264 % Esto es robusto frente a cualquier reordenamiento de índices por el diafragma.
265 % Ref: Chopra (2012), ec. 9.2.1
266 for i = 1:n_dof_6
267     idx_original = gdl_mapping.independientes(gdl_libres(i));
268     tipo_gdl = mod(idx_original - 1, 6) + 1; % 1=Ux,2=Uy,3=Uz,4=Rx,5=Ry,6=Rz
269     nodo_idx = ceil(idx_original / 6); % número de nodo
270     z_nodo = xyz(nodo_idx, 3); % altura del nodo [m]
271     if tipo_gdl == 1 && z_nodo > 0 % Ux en nodo de piso (no base)
272         J_x(i) = 1;
273         idx_Ux = [idx_Ux; i];
274     elseif tipo_gdl == 2 && z_nodo > 0 % Uy en nodo de piso (no base)
275         J_y(i) = 1;
276         idx_Uy = [idx_Uy; i];
277     end
278 end
279
280 % Seleccionar vector de influencia y conjunto de índices según dirección de
    análisis
281 % Ref: Chopra (2012), ec. 9.2.1; NTC-Sismo 2020, Sec. 8.2
282 if strcmpi(dir_analisis, 'Y')
283     J_dom = J_y;
284     idx_dom = idx_Uy;
285     fprintf(' Dirección de análisis: Y (acelerograma 19S-SCT2 Norte-Sur)\n');
286 else
287     J_dom = J_x;
288     idx_dom = idx_Ux;
289     fprintf(' Dirección de análisis: X\n');
290 end
291
292 % 6.1.2 Masa efectiva del modo fundamental - forma matricial exacta
293 % Ref: NTC-Sismo 2020, ec. 6.1.1; Avilés López et al. (2016), ec. 5.1
294 %
295 %  $M^* = \phi(' M J)^2 / \phi(' M \phi)$ 
296 %
297 % Se selecciona el modo con mayor masa efectiva en la dirección de análisis.
298 % En estructuras con acoplamiento torsión-traslación, el primer modo que
    devuelve
299 % eigs puede ser torsional o traslacional en la dirección perpendicular.
300 % Ref: Chopra (2012), Sec. 13.2; Avilés López et al. (2016), Sec. 6.5.1
301 Me_todos = zeros(n_modos, 1);
302 for ii = 1:n_modos
303     phi_ii = phi_br(:, ii);
304     G_ii = phi_ii' * M_red * J_dom;
305     Me_todos(ii) = G_ii^2 / (phi_ii' * M_red * phi_ii);
306 end
307 [~, idx_modo_dom] = max(Me_todos);
308 if idx_modo_dom > 1
309     fprintf(' NOTA: Modo traslacional dominante en %s es el modo %d (T=%0.3f s).\n', ...
310         dir_analisis, idx_modo_dom, T_br(idx_modo_dom));

```

```

311     fprintf(' El modo 1 no es traslacional en %s. Se usa modo %d para el ISE.\n\n'
312           ', ...
313     dir_analisis, idx_mododo_dom);
314 else
315     fprintf(' Modo traslacional dominante en %s: modo 1 (T=%.3f s)\n\n', ...
316           dir_analisis, T_br(1));
317 end
318 % Compatibilidad con código posterior que usa idx_mododo_X en reportes
319 idx_mododo_X = idx_mododo_dom;
320
321 phil = phi_br(:, idx_mododo_dom); % modo traslacional dominante en dir_analisis
322 phil_MJ = phil' * M_red * J_dom; % factor de participación [kg]
323 phil_Mphil = phil' * M_red * phil; % masa generalizada m* [kg]
324 M_e = phil_MJ^2 / phil_Mphil; % masa efectiva modal [kg]
325 W_e = M_e * g; % peso efectivo [N]
326 % (reporte de Me se hace más abajo, junto con Mt, una vez calculados)
327
328 % 6.2 Altura efectiva del modo fundamental
329 %  $H_{\sim} = \Sigma(\phi_m \cdot h_{\sim}) / \Sigma(\phi_m \cdot \sim)$ 
330 % donde h es la altura del nivel n sobre el desplante.
331 % Ref: Avilés López et al. (2016), ec. 6.5.4; NTC-Sismo 2020, Sec. A4
332 %
333 % Para el método estático se permite  $H_e = 0.7 \cdot H$  (NTC-Sismo 2020, Sec. A4).
334 % Aquí se calcula explícitamente desde la forma modal para mayor rigor.
335
336 % 6.2.1 Alturas de los GDLs traslacionales  $U_x$ 
337 % Se recupera el índice original (antes del diafragma) para calcular
338 % correctamente el número de nodo y su coordenada z.
339 h_nodos_dom = zeros(length(idx_dom), 1);
340 for i = 1:length(idx_dom)
341     gdl_red = gdl_libres(idx_dom(i)); % índice en espacio reducido
342     gdl_orig = gdl_mapping.independientes(gdl_red); % índice original (6*nNodos)
343     nodo_idx = ceil(gdl_orig / 6); % número de nodo
344     h_nodos_dom(i) = xyz(nodo_idx, 3); % altura z [m]
345 end
346
347 % 6.2.2 Altura efectiva - forma matricial
348 % Se construye un vector h_vec con la altura de cada GDL (altura del nodo
349 % para GDLs  $U_x$ , cero para el resto). Entonces:
350 %
351 %  $H_{\sim} = \phi^T(\sim M h_{\sim}) / \phi^T(\sim M J_x)$ 
352 %
353 % Esta forma es robusta frente a M no diagonal (post-diafragma) y es
354 % consistente con la definición de Avilés López (2016) ec. 6.5.4 en
355 % forma continua discretizada.
356 % Ref: Avilés López et al. (2016), ec. 6.5.4
357
358 h_vec = zeros(n_dof_6, 1);
359 for i = 1:length(idx_dom)
360     h_vec(idx_dom(i)) = h_nodos_dom(i);
361 end
362
363 phil_Mh = phil' * M_red * h_vec; % (1xn)·(n×n)·(n×1) -> escalar [kg·m]
364 % phil_MJ ya calculado arriba: phil' * M_red * J_dom -> escalar [kg]

```

```

365 He_modal = abs(phi1_Mh / phi1_MJ); % [m] abs() protege contra signo del modo
366
367 % 6.2.3 Comparación con aproximación He = 0.7*H
368 H_total = nNiv * h;
369 He_aprox = 0.7 * H_total;
370
371 % Masa diagonal GDLs dominantes (para referencia en validación de masa total
solamente)
372 m_diag_dom = full(diag(M_red(idx_dom, idx_dom)));
373 M_total_dom = sum(m_diag_dom);
374
375 % Reporte consolidado de masas
376 fprintf(' Masa efectiva modal (modo %d): M* = %.1f ton\n', idx_modos_dom, M_e/1e3
);
377 fprintf(' Peso efectivo modal: W* = %.1f kN\n', W_e/1e3);
378 fprintf(' Masa total traslacional: Mt = %.1f ton\n', M_total_dom/1e3);
379 fprintf(' Razón M*/Mt = %.3f (rango típico 0.60-0.85)\n\n', M_e/M_total_dom);
380
381 fprintf(' Altura efectiva (modal): He = %.2f m\n', He_modal);
382 fprintf(' Altura efectiva (aprox 0.7H): He = %.2f m ... referencia NTC estático\
n', He_aprox);
383 fprintf(' Se usará He modal para mayor rigor.\n\n');
384 He = He_modal;
385
386 %-----
387 % LAZO ITERATIVO - Impedancias evaluadas en omega1~ = 2*pi/T~
388 %
389 % Las impedancias Kh, Kr, Ch, Cr son funciones de la frecuencia de
390 % excitación (Gazetas 1991; Avilés López 2016, Sec. 6.3.2). El
391 % procedimiento riguroso de NTC-Sismo 2020 Sec. 8.2.3 requiere evaluar
392 % estos parámetros en omega1~ = 2*pi/T~, que es la incógnita del problema.
393 %
394 % Algoritmo de punto fijo:
395 % 1. Inicializar: T~ = Te (base rígida)
396 % 2. Evaluar Ch(omega1), Cr(omega1) en omega1 = 2*pi/T~
397 % 3. Calcular Th, Tr con ec. 8.2.8-8.2.9
398 % 4. Calcular T~_nuevo = sqrt(Te^2+Th^2+Tr^2) - ec. 8.2.7
399 % 5. Verificar |T~_nuevo - T~| / T~ < tolerancia -> salir
400 % 6. Si no: T~ = T~_nuevo, ir a 2
401 %
402 % Convergencia: típicamente 3-6 iteraciones en Zona III.
403 % Ref: NTC-Sismo 2020, Sec. 8.2.3 (comentario); Avilés López et al.
404 % (2016), Sec. 6.5.2.1
405 %-----
406
407 Te = T_br(idx_modos_dom); % periodo base rígida - modo traslacional dominante [s]
408 xi_e = 0.05; % amortiguamiento estructural
409 tol_iter = 1e-6; % tolerancia relativa de convergencia
410 max_iter = 50; % límite de iteraciones (protección)
411
412 % Tabla de convergencia para reporte
413 fprintf(' ITERACIÓN - Impedancias evaluadas en omega = 2*pi/T~\n');
414 fprintf(' +-----+\n');
415 fprintf(' | Iter | T~ (s) | T1 (s) | Ts (s) | Error |\n');
416 fprintf(' +-----+\n');

```

```

417
418 T_tilde = Te; % valor inicial: primera iteración en omega = 2*pi/T~
419
420 for iter = 1:max_iter
421
422     omega_eff = 2*pi / T_tilde; % frecuencia efectiva actual [rad/s]
423
424     % Coeficientes dinámicos evaluados en omegal
425     % Para semiespacio uniforme kh ≈1 (Gazetas 1991), por lo que
426     % Kh(omegal) ≈Kh_static. Lo que varía con la frecuencia es Ch y Cr.
427     Ch_iter = Ch_fun(omega_eff); % [N*s/m]
428     Cr_iter = Cr_fun(omega_eff); % [N*m*s/rad]
429
430     % Periodos parciales (NTC-Sismo 2020, ecs. 8.2.8 y 8.2.9)
431     Th_sq = (2*pi)^2 * W_e / (g * Kh_static);
432     Tr_sq = (2*pi)^2 * W_e * (He + D_cim)^2 / (g * Kr_static);
433     Th = sqrt(Th_sq);
434     Tr = sqrt(Tr_sq);
435
436     % Periodo efectivo - ec. 8.2.7
437     T_tilde_new = sqrt(Te^2 + Th_sq + Tr_sq);
438
439     % Error relativo
440     err_rel = abs(T_tilde_new - T_tilde) / T_tilde;
441
442     fprintf(' | %3d | %.6f | %.6f | %.6f | %.2e |\n', ...
443             iter, T_tilde_new, Th, Tr, err_rel);
444
445     T_tilde = T_tilde_new;
446
447     if err_rel < tol_iter
448         break
449     end
450
451     if iter == max_iter
452         warning('PASO 6: el lazo iterativo no convergió en %d iteraciones.',
453                 max_iter);
454     end
455 end
456
457 fprintf(' +-----+\n');
458 fprintf(' [OK] Convergencia en %d iteración(es) (tol = %.0e)\n\n', iter,
459         tol_iter);
460
461 % Valores finales convergidos
462 T_tilde_sq = T_tilde^2;
463 dT_abs = T_tilde - Te;
464 dT_pct = 100 * dT_abs / Te;
465
466 fprintf(' +-----+\n');
467 fprintf(' T~^2 = Te^2 + Th^2 + Tr^2 (solución iterada)\n');
468 fprintf(' +-----+\n');
469 fprintf(' Te (base rígida): %.4f s\n', Te);
470 fprintf(' Th (traslación): %.4f s\n', Th);
471 fprintf(' Tr (cabeceo): %.4f s\n', Tr);

```

```

470 fprintf(' T~ (base flexible): %.4f s\n', T_tilde);
471 fprintf(' dT = +%.4f s (dT/Te = +%.1f%%)\n', dT_abs, dT_pct);
472 fprintf(' +-----+\n');
473
474 % Contribuciones porcentuales al incremento
475 contrib_Th_pct = 100 * Th_sq / (T_tilde_sq - Te^2 + eps);
476 contrib_Tr_pct = 100 * Tr_sq / (T_tilde_sq - Te^2 + eps);
477 fprintf(' Contribución al incremento T~^2 - Te^2:\n');
478 fprintf(' Traslación (Th^2): %.1f%%\n', contrib_Th_pct);
479 fprintf(' Cabeceo (Tr^2): %.1f%%\n', contrib_Tr_pct);
480 fprintf('\n');
481
482 fprintf(' Rango esperado Zona III: +20%% a +55%%\n');
483 fprintf(' Refs: Avilés et al. (2016); Mayoral et al. (2019);\n');
484 fprintf(' Quinde & Reinoso (2020)\n\n');
485
486 if dT_pct < 20
487     fprintf(' *** NOTA: ΔT < 20%%. Verificar parámetros del suelo.\n\n');
488 elseif dT_pct > 55
489     fprintf(' *** NOTA: ΔT > 55%%. Resultado alto -verificar Kr y He.\n\n');
490 else
491     fprintf(' +-----+\n');
492 end
493
494 % Amortiguamiento efectivo - NTC-Sismo 2020, ecs. 8.2.10-8.2.12
495 % Evaluado con Ch y Cr en la frecuencia convergida omega1~ = 2*pi/T~
496 %
497 % ξ~h = Ch(omega1~)·T~ / (2*pi·Kh) ... ec. 8.2.11
498 % ξ~r = Cr(omega1~)·T~ / (2*pi·Kr) ... ec. 8.2.12
499 % ξ~ = δ[e·Te^3 + ξ~h·Th^2·T~ + ξ~r·Tr^2·T~] / T~^3 ... ec. 8.2.10
500
501 delta_h = Ch_iter * T_tilde / (2*pi * Kh_static);
502 delta_r = Cr_iter * T_tilde / (2*pi * Kr_static);
503
504 xi_tilde = (xi_e * Te^3 + delta_h * Th_sq * T_tilde + delta_r * Tr_sq * T_tilde)
505     ...
506     / T_tilde^3;
507
508 % Restricción NTC: mínimo 5% cuando no se incluye interacción cinemática
509 % (NTC-Sismo 2020, Sec. 8.2.3, comentario)
510 xi_tilde = max(xi_tilde, 0.05);
511
512 fprintf(' Amortiguamiento efectivo (omega~ convergida):\n');
513 fprintf(' xi_e (estructura): %.4f (%.1f%%)\n', xi_e, 100*xi_e);
514 fprintf(' xi~h (traslación): %.4f (%.1f%%)\n', delta_h, 100*delta_h);
515 fprintf(' xi~r (cabeceo): %.4f (%.1f%%)\n', delta_r, 100*delta_r);
516 fprintf(' xi~ (sistema): %.4f (%.1f%%)\n', xi_tilde, 100*xi_tilde);
517 fprintf(' Ch (convergado): %.4e N*s/m\n', Ch_iter);
518 fprintf(' Cr (convergado): %.4e N*m*s/rad\n\n', Cr_iter);
519
520 % Aplicar T~ al modo traslacional dominante; modos restantes sin cambio
521 % Ref: NTC-Sismo 2020, Sec. 8.2; Avilés López et al. (2016), Sec. 6.5.3
522 T_bf = T_br;
523 T_bf(idx_modos_dom) = T_tilde;
524 omega_bf = 2*pi ./ T_bf;

```

```

524 f_bf = 1 ./ T_bf;
525
526 % Variables para el resumen final
527 factor_Kh = Th_sq / Te^2;
528 factor_Kr = Tr_sq / Te^2;
529 factor_total = T_tilde_sq / Te^2;
530
531 % Tabla comparativa de periodos
532 fprintf(' Comparación de periodos (modo 1 modificado por ISE):\n');
533 fprintf(' +-----+\n');
534 fprintf(' | Modo | T_br (s) | T_bf (s) | dT (s) | dT/T (%%) |\n');
535 fprintf(' +-----+\n');
536 for i = 1:min(3, n_modos)
537     dT_i = T_bf(i) - T_br(i);
538     dT_pct_i = 100 * dT_i / T_br(i);
539     if i == 1
540         fprintf(' | %d | %.4f | %.4f | +%.4f | +%.1f%% | <- ISE\n', ...
541             i, T_br(i), T_bf(i), dT_i, dT_pct_i);
542     else
543         fprintf(' | %d | %.4f | %.4f | 0.0000 | 0.0%% |\n', ...
544             i, T_br(i), T_bf(i));
545     end
546 end
547 fprintf(' +-----+\n');
548 fprintf(' NOTA: Los modos superiores no se modifican por ISE.\n');
549 fprintf(' Ref: NTC-Sismo (2020), Sec. 8.2; Avilés López et al. (2016), Sec.
550     6.5.3\n\n');
551 %% 7. Cargar acelerograma
552 fprintf('PASO 7: Acelerograma 19S-2017\n');
553 fprintf('-----\n');
554
555 ag_data = load('19S_SCT2.txt');
556 ag_data = ag_data / 100; % cm/s^2 -> m/s^2
557 dt = 0.01;
558 n_steps = length(ag_data);
559 t_vec = (0:n_steps-1)' * dt;
560
561 fprintf(' Duración: %.1f s, a_max=%.2f m/s^2\n', t_vec(end), max(abs(ag_data)));
562 fprintf('\n');
563
564 %% 8. Análisis tiempo-historia (BASE RÍGIDA) - Excitación X e Y
565 fprintf('PASO 8: Análisis tiempo-historia (base rígida)\n');
566 fprintf('-----\n');
567 fprintf(' Integrando %d pasos con Newmark-beta (aceleración promedio)... \n',
568     n_steps);
569 fprintf(' Excitación: X e Y simultáneos (mismo acelerograma, en fase)\n\n');
570
571 % 8.1 Parámetros del integrador de Newmark
572 % beta=1/4, gamma=1/2: método de aceleración promedio constante.
573 % Incondicionalmente estable; no introduce disipación numérica artificial.
574 % Ref: Chopra (2012), Sec. 5.2; Bathe (1996)
575 beta_n = 1/4;

```

```

575 gamma_n = 1/2;
576 xi_est = 0.05; % Amortiguamiento viscoso equivalente (5% crítico)
577
578 % 8.2 Amortiguamiento de Rayleigh
579 % C = alpha_R * M_red + beta_R * K
580 % Calibrado para xi=5% en modos 1 y 3 (dominantes en X e Y)
581 % Ref: Chopra (2012), Sec. 11.4
582 %
583 % REGULARIZACIÓN DE M_red:
584 % M_red tiene GDLs con masa exactamente cero (rotaciones y Uz).
585 % Esto produce una M singular que desestabiliza Newmark aunque beta=1/4
586 % sea teóricamente incondicional. Se añade masa ficticia mínima en esos
587 % GDLs para regularizar M sin afectar la respuesta traslacional.
588 % El factor epsilon=1e-6 produce masa ficticia de orden 7.5e-3 kg,
589 % despreciable frente a la masa estructural mínima de 7530 kg (factor 1e6).
590 % Ref: Hughes (1987), Sec. 9.3.
591 M_diag_reg = full(diag(M_red));
592 m_min_nz = min(M_diag_reg(M_diag_reg > 0));
593 epsilon_m = 1e-6;
594 idx_cero = find(M_diag_reg == 0);
595 M_red_reg = M_red + sparse(idx_cero, idx_cero, ...
596     epsilon_m * m_min_nz * ones(length(idx_cero),1), ...
597     size(M_red,1), size(M_red,2));
598 fprintf(' Regularización M_red: %d GDLs con masa=0, masa ficticia=%.2e kg\n',
599     ...
600     length(idx_cero), epsilon_m * m_min_nz);
601
602 w1 = omega_br(1);
603 w3 = omega_br(min(3, length(omega_br)));
604 alpha_R = 2 * xi_est * w1 * w3 / (w1 + w3);
605 beta_R = 2 * xi_est / (w1 + w3);
606 C_total = alpha_R * M_red_reg + beta_R * K_red;
607
608 fprintf(' Amortiguamiento Rayleigh: alpha=%.4e, beta=%.4e\n', alpha_R, beta_R);
609 fprintf(' (Calibrado para xi=5%% en modos 1 y 3)\n\n');
610
611 % 8.3 Vectores de influencia
612 % La ecuación de movimiento bajo excitación de suelo es:
613 %  $M \cdot \ddot{u} + C \cdot \dot{u} + K \cdot u = -M \cdot \iota^1_x \cdot a_{g_x}(t) - M \cdot \iota^1_y \cdot a_{g_y}(t)$ 
614 % donde  $\iota^1_x$  e  $\iota^1_y$  son los vectores de influencia que proyectan la
615 % aceleración del suelo sobre los GDLs de la estructura.
616 %  $\iota^1_x(i) = 1$  si el GDL i es traslacional en X, 0 en caso contrario.
617 %  $\iota^1_y(i) = 1$  si el GDL i es traslacional en Y, 0 en caso contrario.
618 %
619 % CORRECTO: Vector de unos solo en los GDLs traslacionales de la
620 % dirección de análisis.
621 % Ref: Chopra (2012), ec. 9.2.1; Clough & Penzien (1993), Sec. 26.4
622
623 n_dof = size(K_red, 1);
624 iota_x = zeros(n_dof, 1);
625 iota_y = zeros(n_dof, 1);
626
627 for i = 1:length(gdl_libres)
628     idx_original = gdl_mapping.independientes(gdl_libres(i));

```

```

629     tipo_gdl = mod(idx_original - 1, 6) + 1; % 1=Ux,2=Uy,3=Uz,4=Rx,5=Ry,6=Rz
630     nodo_idx = ceil(idx_original / 6);
631     z_nodo = xyz(nodo_idx, 3);
632     if tipo_gdl == 1 && z_nodo > 0 % Ux en nodo de piso (no base)
633         iota_x(i) = 1;
634     elseif tipo_gdl == 2 && z_nodo > 0 % Uy en nodo de piso (no base)
635         iota_y(i) = 1;
636     end
637 end
638
639 n_Ux = sum(iota_x);
640 n_Uy = sum(iota_y);
641 fprintf(' GDLs en vector iota_x (Ux activos): %d\n', n_Ux);
642 fprintf(' GDLs en vector iota_y (Uy activos): %d\n', n_Uy);
643 fprintf(' GDLs restantes (rotaciones, Uz): no excitados\n\n');
644
645 % 8.4 Inicialización de vectores de estado
646 u = zeros(n_dof, n_steps);
647 v = zeros(n_dof, n_steps);
648 a = zeros(n_dof, n_steps);
649
650 % 8.5 Condición inicial (t=0)
651 % La fuerza efectiva incluye contribuciones de X e Y simultáneamente.
652 F_ext_0 = -M_red_reg * (iota_x * ag_data(1) + iota_y * ag_data(1));
653 % Condición inicial de reposo (estructura en reposo antes del sismo).
654 % M_red_reg es no singular y permite resolver el sistema inicial.
655 % Ref: Chopra (2012), Sec. 5.2
656 a(:,1) = M_red_reg \ (F_ext_0 - C_total*v(:,1) - K_red*u(:,1));
657
658 % 8.6 Constantes del algoritmo de Newmark
659 a0_n = 1 / (beta_n * dt^2);
660 a1_n = gamma_n / (beta_n * dt);
661 a2_n = 1 / (beta_n * dt);
662 a3_n = 1/(2*beta_n) - 1;
663 a4_n = gamma_n/beta_n - 1;
664 a5_n = dt/2 * (gamma_n/beta_n - 2);
665 a6_n = dt * (1 - gamma_n);
666 a7_n = gamma_n * dt;
667
668 % Matriz de rigidez efectiva (constante para beta=1/4, gamma=1/2)
669 K_eff = K_red + a0_n*M_red_reg + a1_n*C_total;
670
671 % Regularización mínima para evitar singularidad numérica residual
672 epsilon_reg = 1e-8 * max(diag(K_eff));
673 K_eff = K_eff + epsilon_reg * speye(size(K_eff));
674
675 fprintf(' Factorizando K_eff (LU)... \n');
676 [L_lu, U_lu, P_lu] = lu(K_eff);
677 fprintf(' ----- \n');
678
679 fprintf(' Integrando');
680 tic;
681 prog_ant = 0;
682
683 % 8.7 Bucle de integración temporal

```

```

684 for k = 2:n_steps
685     % Vector de fuerza efectiva con excitación X e Y simultánea
686     F_sismo = -M_red_reg * (iota_x * ag_data(k) + iota_y * ag_data(k));
687
688     p_eff = F_sismo + ...
689         M_red_reg * (a0_n*u(:,k-1) + a2_n*v(:,k-1) + a3_n*a(:,k-1)) + ...
690         C_total * (a1_n*u(:,k-1) + a4_n*v(:,k-1) + a5_n*a(:,k-1));
691
692     u(:,k) = U_lu \ (L_lu \ (P_lu * p_eff));
693     a(:,k) = a0_n*(u(:,k) - u(:,k-1)) - a2_n*v(:,k-1) - a3_n*a(:,k-1);
694     v(:,k) = v(:,k-1) + a6_n*a(:,k-1) + a7_n*a(:,k);
695
696     prog = floor(100*k/n_steps);
697     if prog > prog_ant && mod(prog,10)==0
698         fprintf(' %d%%', prog);
699         prog_ant = prog;
700     end
701 end
702
703 t_calc = toc;
704 fprintf(' [OK] (%.1f s)\n\n', t_calc);
705
706 % 8.8 Desplazamiento máximo resultante (GDLs traslacionales X e Y)
707 u_max_global = 0;
708 for i = 1:length(gdl_libres)
709     idx_original = gdl_mapping.independientes(gdl_libres(i));
710     tipo_gdl = mod(idx_original - 1, 6) + 1;
711     nodo_idx = ceil(idx_original / 6);
712     z_nodo = xyz(nodo_idx, 3);
713     if (tipo_gdl == 1 || tipo_gdl == 2) && z_nodo > 0 % Ux o Uy en nodo de piso
714         u_max_gdl = max(abs(u(i,:)));
715         if u_max_gdl > u_max_global
716             u_max_global = u_max_gdl;
717         end
718     end
719 end
720 u_max = u_max_global;
721
722 fprintf(' u_max horizontal (envolvente X-Y) = %.4f m (%.2f cm)\n', ...
723         u_max, u_max*100);
724 fprintf('\n');
725
726 %% 9. Gráficas
727 fprintf('PASO 9: Gráficas\n');
728 fprintf('-----\n');
729
730 carpeta = 'resultados_ise';
731 if ~exist(carpeta, 'dir'), mkdir(carpeta); end
732
733 % 9.1 Gráfica 1: Periodos
734 figure('Visible', 'off', 'Position', [100,100,800,500]);
735 n_plot = min(3, n_modos);
736 datos_bar = [T_br(1:n_plot), T_bf(1:n_plot)];
737 b = bar(1:n_plot, datos_bar);

```

```

738 b(1).FaceColor = [0.2 0.4 0.8];
739 b(2).FaceColor = [0.8 0.2 0.2];
740 xlabel('Modo'); ylabel('Periodo (s)');
741 title('Comparación: Base Rígida vs Base Flexible');
742 legend('Base Rígida', 'Base Flexible');
743 grid on;
744 saveas(gcf, fullfile(carpeta, 'periodos.png'));
745 close(gcf);
746 fprintf(' Gráfica guardada: %s/periodos.png\n', carpeta);
747
748 fprintf('\n');
749 fprintf('ANÁLISIS COMPLETADO\n');
750 fprintf('\n');
751
752 % % fprintf(' Resultados: %s\n', carpeta);
753 % % fprintf(' Guardando workspace...\n');
754 % % save(fullfile(carpeta, 'workspace_ise.mat'), '-v7.3');
755 % % fprintf(' +-----+\n');
756
757 fprintf(' RESUMEN ANÁLISIS ISE:\n');
758 fprintf(' +-----+\n');
759 fprintf(' Dirección de análisis: %s\n', dir_analisis);
760 fprintf(' Modo traslacional dominante en %s: modo %d\n', dir_analisis,
       idx_modos_dom);
761 fprintf(' Periodo fundamental (base rígida): T1 = %.3f s\n', Te);
762 fprintf(' Periodo fundamental (base flexible): T~ = %.3f s\n', T_tilde);
763 fprintf(' Incremento total por ISE: dT/T1 = +%.1f%%\n', ...
       100*(T_tilde - Te)/Te);
764 fprintf(' Contribución traslacional (Kh): %.1f%%\n', ...
       100*factor_Kh/factor_total);
765 fprintf(' Contribución rotacional (Kr / cabeceo): %.1f%%\n', ...
       100*factor_Kr/factor_total);
766 fprintf(' Desplazamiento máximo horizontal: u_max = %.2f cm\n', u_max*100);
767 fprintf(' +-----+\n');
768 fprintf(' Rango típico Zona III (literatura): +20%% a +55%%\n');
769 fprintf(' Refs: Avilés et al. 2016; Mayoral et al. 2019;\n');
770 fprintf(' Quinde & Reinoso 2020\n');
771 fprintf('\n');
772 fprintf('PASO SIGUIENTE: Validar con SAP2000\n\n');

```