

DESARROLLO E IMPLEMENTACIÓN DE UNA INTERFAZ GRÁFICA PARA EL PROGRAMA DE ELEMENTOS FINITOS FEM

LUIS FELIPE GUERRERO, Ing.

Ingeniero Civil, Universidad del Valle, Colombia. Grupo de Investigación en Ingeniería Sísmica, Ingeniería Eólica y Estructuras Inteligentes, G-7. Estudiante de Maestría, Escuela de Ingeniería Civil y Geomática, Universidad del Valle, Cali, Colombia.

felipe.guerrero@correounivalle.edu.co

DANIEL GÓMEZ PIZANO, M.Sc

Ingeniero Civil, Magister en Ingeniería Civil, Universidad del Valle, Colombia. Grupo de Investigación en Ingeniería Sísmica, Ingeniería Eólica y Estructuras Inteligentes, G-7. Profesor Asistente, Escuela de Ingeniería Civil y Geomática, Universidad del Valle, Cali, Colombia.

daniel.gomez@correounivalle.edu.co

PETER THOMSON, Ph.D

Ingeniero Aeroespacial, Universidad de Minnesota, USA. Director Grupo de Investigación en Ingeniería Sísmica, Ingeniería Eólica y Estructuras Inteligentes, G-7. Profesor Titular, Escuela de Ingeniería Civil y Geomática, Universidad del Valle, Cali, Colombia.

pethomso@univalle.edu.co

RESUMEN

La aplicación FEM es una caja de herramientas para el análisis de estructuras por el método de elementos finitos creado en el programa matemático MATLAB de libre distribución. Se desarrolló para que los estudiantes y profesores tengan una herramienta útil para comprender el análisis de estructuras, permitiendo entre otros visualizar la matriz de rigidez global, la matriz de masa diagonal o consistente, etc. a diferencia de los programas comerciales, los cuales son cajas negras que no permiten la manipulación de algunos datos. Sin embargo, realizar el modelo de una estructura en la aplicación FEM era dispendioso debido a que no se contaba con una interfaz gráfica de usuario para la inserción de datos ni para la visualización de los resultados que facilitara el uso del programa y la interpretación de estos. El presente artículo describe el desarrollo e implementación de una interfaz gráfica que permite mejorar la funcionalidad y facilidad de uso respecto a la versión preexistente del programa de elementos finitos FEM para la modelación de estructuras, basado en el lenguaje de programación de MATLAB. Con esta interfaz gráfica se mejoró el ingreso de datos al programa haciendo este proceso más fácil e intuitivo.

Palabras clave: Programación orientada a objetos, Interfaz gráfica, Elementos finitos, Modelación de estructuras.

ABSTRACT

The FEM application is a toolbox for the analysis of structures by the finite element method developed in the mathematical program MATLAB and developed for students and teachers were a useful tool for understanding the structural analysis, allowing among other matrix display global stiffness, the diagonal mass matrix or consistent, and so on unlike commercial programs, which are black boxes that do not allow manipulation of some data. However, creating the model of a structure in the FEM application was time consuming because there were no graphical user interface for data entry and for displaying the results to facilitate the use of the program and the interpretation of these. This article results from an investigation that was developed and implementing a graphical interface that improves the pre-and post-processing of pre-existing version of the FEM finite element program for modeling of structures, based on the MATLAB programming language.

Keywords: Object-oriented programming, GUI, Finite elements, modeling of structures.

1.INTRODUCCIÓN

En el campo de la ingeniería los investigadores invierten gran parte de su tiempo validando nuevas metodologías mediante experimentos físicos y analíticos. Debido al desarrollo de la computación en los últimos años, los experimentos numéricos son cada vez más comunes y constituyen el fundamento para demostrar la validez de las nuevas metodologías o teorías que están desarrollando. Generalmente en Ingeniería Estructural se utilizan modelos basados en el método de los Elementos Finitos, ya que este tipo de metodologías han demostrado ser apropiados para estructuras civiles.

Existe una amplia gama de software diseñados para el análisis de las estructuras por medio del método de elementos finitos. Entre ellos se encuentran aquellos que están específicamente diseñados para la resolución de problemas de ingeniería geotécnica, los utilizados para el análisis de esfuerzo de estructuras en dos dimensiones, aquellos especializados en el análisis de estructuras de ingeniería civil, tales como estructuras y puentes; con análisis lineales y no lineales de estructuras bidimensionales y tridimensionales, programas como el ETABS y el SAP2000, realizan análisis computacionales más complejos facilitando el cálculo y el diseño de estructuras.

Sin embargo, estos programas presentan limitaciones cuando son utilizados en el ámbito académico e investigativo, en tanto no permiten acceder a datos relacionados con las propiedades de las estructuras, como las matrices de rigidez y masa, ni a los procesos internos de los cálculos. Además la mayoría presenta la dificultad de fusionarlos con nuevos códigos desarrollados por los investigadores.

Este artículo describe el desarrollo y la implementación de una interfaz gráfica en Matlab, para el programa de elementos finitos FEM que le permita a los investigadores una fácil implementación de experimentos numéricos para validar sus metodologías y que además este programa sirva de apoyo a los cursos avanzados del área de la mecánica de sólidos de la Ingeniería Civil, permitiendo modelar edificios o puentes. Así, con el diseño de la interfaz gráfica, el *FEM* amplía su utilidad como herramienta de análisis mediante el método de elementos finitos, contribuyendo con su practicidad en el área académica, de investigación y en los diversos desarrollos profesionales.

2. MARCO TEÓRICO

2.1 Elementos Finitos

El término elemento finito resume el concepto básico del método, la transformación de un sistema físico, con un número infinito de incógnitas, a uno que tiene un número finito de incógnitas relacionadas entre sí por elementos de un tamaño finito. El Método de Elementos Finitos permite obtener una solución numérica aproximada sobre una estructura, en la que están definidas ciertas ecuaciones diferenciales que caracterizan el comportamiento físico de la estructura.

Para ello trabaja discretizando la estructura en elementos de forma variada (pueden ser superficies, volúmenes o barras). La solución obtenida ahora es sólo aproximada, coincidiendo con la solución exacta sólo en un número finito de puntos llamados nodos, los cuales son una localización en el espacio de un punto en el que se considera que existen ciertos grados de libertad (desplazamientos, temperaturas, etc.) y acciones (fuerzas, condiciones de contorno) de la estructura. En el resto de puntos que no son nodos, la solución aproximada se obtiene extrapolando a partir de los resultados obtenidos para los nodos, lo cual hace que la solución sea sólo aproximada debido a ese último paso.

El Método de Elementos Finitos convierte un problema definido en términos de ecuaciones diferenciales en un problema en forma matricial que proporciona el resultado correcto para un número de finito de puntos y extrapola posteriormente la solución al resto del dominio, resultando finalmente sólo una solución aproximada. El conjunto de puntos donde la solución es exacta se denomina nodos. Dicho conjunto de nodos forman una red, denominada malla. Cada uno de los retículos contenidos en dicha malla es un "elemento finito".

2.2 Programación Orientada a objetos

La programación orientada a objetos (POO) es un paradigma de programación que usa de manera estructurada objetos como un conjunto complejo de datos y sus interacciones, para diseñar aplicaciones y programas informáticos. El enfoque de la programación orientada a objetos mejora su capacidad para administrar un software complejo, mediante la aplicación de sistemas de clasificación y los patrones de diseño a la programación. El uso de la programación orientada a objetos es especialmente importante en el desarrollo y mantenimiento de aplicaciones de gran tamaño y estructuras de datos.

La idea fundamental en los lenguajes orientados a objetos es combinar en una única unidad tanto datos como funciones que operan sobre estos datos, a tal unidad se le denomina objeto. Estos poseen propiedades o atributos importantes y forman parte de una organización jerárquica o de otro tipo. Estos objetos contienen información que permite definirlos e identificarlos frente a otros objetos.

Un conjunto de objetos con características comunes se describe como una clase. Los objetos son instancias específicas de una clase. Los valores que figuran en las propiedades de un objeto es lo que hace un objeto diferente de otros objetos de la misma clase ^[1].

Los métodos son funciones definidas por las clases, son los que ponen en práctica comportamientos del objeto que son comunes a todos los objetos de una clase y además definen la interacción entre objetos. Una clase requiere de los métodos para poder acceder y usar los atributos de un objeto, de esta manera se realiza una programación estructurada orientada a objetos. Las definiciones de clase describen cómo los objetos de la clase crean los datos que contienen los objetos y cómo se puede manipular estos datos.

Entre las características principales de la programación orientada a objetos encontramos la herencia, abstracción, polimorfismo y encapsulamiento ^[2]. La herencia es la facilidad que unas clases tienen para derivar de otras y aprovechar su forma de trabajar sin tener que reescribir código. La clase derivada o clase hija, recibe todos los atributos y métodos de las clases de la cual derivan de la clase

madre, según tenga la clase madre registrado el acceso a los atributos y métodos como público o privado. La abstracción denota las características esenciales de un objeto que lo distinguen de cualquier otro tipo de objetos, donde se capturan sus comportamientos, este proceso permite seleccionar las características relevantes.

El polimorfismo permite comportamientos diferentes asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento o función correspondiente al objeto que se esté usando. Dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos y la invocación de una función en una referencia producirá el comportamiento correcto para el tipo del objeto referenciado. Los métodos de una clase pueden mantener el mismo nombre pero hacer cosas distintas según los parámetros que reciban. La encapsulación es el proceso de reunir a todos los elementos de una abstracción que constituyen su estructura y comportamiento. La abstracción y la encapsulación son conceptos complementarios: la abstracción se centra en el comportamiento observable de un objeto, mientras que la encapsulación se centra en la implementación que da lugar a este comportamiento. [3]

Para representar las clases de un programa orientado a objetos se utiliza diagramas de clase, el cual es un tipo de diagrama estático que describe la estructura de un programa mostrando sus clases, atributos y las relaciones entre ellos. La notación corresponde a un rectángulo dividido en tres partes: nombre de la clase, atributos y operaciones.

3. DESCRIPCIÓN DEL FEM

FEM es un programa de elementos finitos creado para facilitar la modelación de estructuras, haciendo uso de la programación orientada a objetos de MATLAB. Inicialmente estaba concebido como una caja de herramientas, de manera global y en perspectiva, con la jerarquía de objetos mostrada en la figura 1 y con capacidad para realizar varios tipos de análisis como análisis estático lineal y análisis modal.

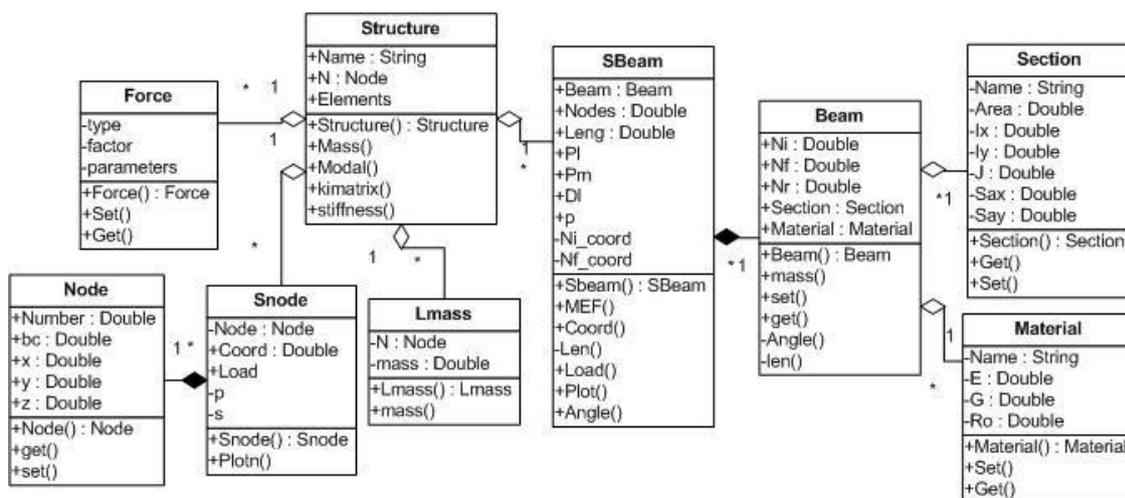


Figura 1. Jerarquía de objetos

De acuerdo con la jerarquía planteada, los objetos que pertenecen a la clase *Structure* son los más importantes y están conformados por objetos de otras clases.

Esta clase es responsable de la gestión de los elementos de las demás clases, de la solución de las ecuaciones del sistema y el cálculo de las propiedades de la estructura, como de la matriz de masa y las frecuencias modales con los métodos de la clase.

Los objetos de la clase *Node* son responsables de guardar las coordenadas de los nodos y las restricciones de los grados de libertad. El objeto de la clase *Node* es un atributo de la clase *Snode*. La clase *Snode* es responsable de asignar las cargas a los nodos y graficarlos.

Los objetos de la clase *Material* contiene las propiedades de los materiales que componen los elementos. Sus atributos son: el módulo de elasticidad, el módulo de cortante y la densidad de masa.

Los objetos de la clase *Section* contienen las propiedades de las secciones de elementos. Sus atributos son: el área, la inercia con respecto al eje X y al eje Y, la inercia rotacional y el módulo de sección del eje X y el eje Y.

Los objetos de la clase *Beam* son responsables de guardar los nodos entre los cuales está definido el elemento viga-columna y tiene como atributo un objeto de la clase *Material* y un objeto de la clase *Section* que definen las propiedades del elemento frame. El elemento de la clase *Beam* es un atributo de la clase *Sbeam*. La clase *Sbeam* se encarga de aplicar las fuerzas a los elementos, orientar la sección transversal de ellos, graficar los elementos y calcular y graficar las fuerzas internas de los elementos.

3.1 Interfaz gráfica

El programa FEM cuenta con una interfaz gráfica de usuario (Figura 2) muy sencilla de usar. Esta interfaz incluye la ventana principal, barra de título, barra de menús, barra de herramientas y ventana de visualización. A continuación se describen cada uno de estos elementos.

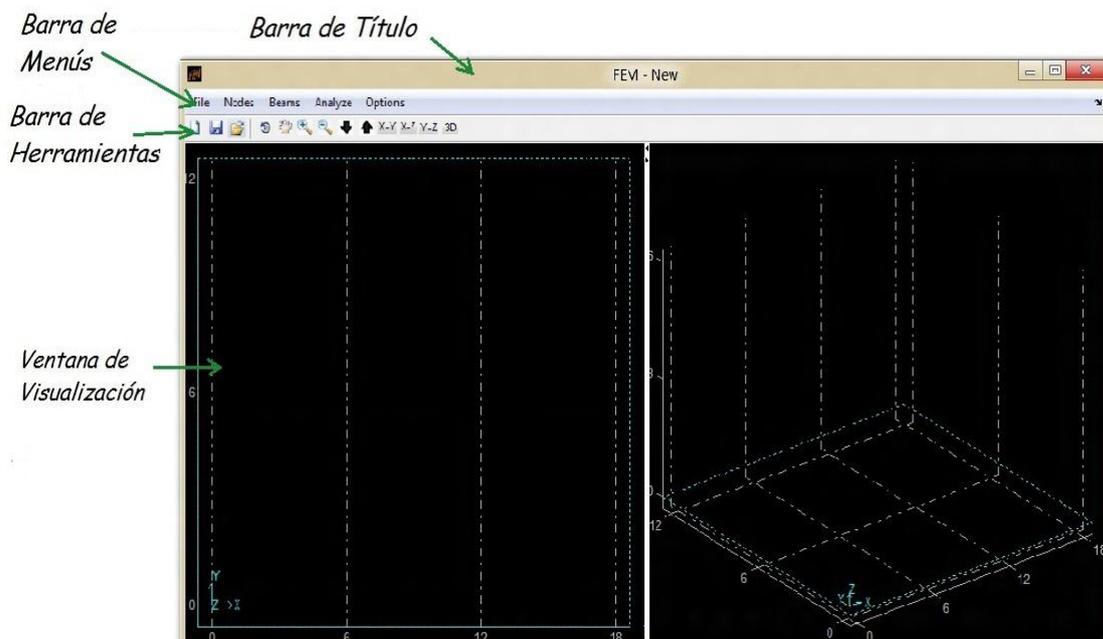


Figura 2. Ventana principal del programa FEM

La ventana principal se puede maximizar, minimizar, cambiar de tamaño, mover o cerrar con las operaciones estándar de Windows. Y contiene los siguientes elementos:

- Barra de título. La barra de título contiene el nombre del programa y del modelo.
- Barra de menús. La barra de menús contiene todos los menús del programa.
- Barra de herramientas. La barra de herramientas se compone de botones que proporcionan acceso a las herramientas de visualización. Manteniendo el cursor sobre un botón de la barra de herramientas podrá ver una corta descripción del funcionamiento del botón.
- Ventana de visualización. La ventana de visualización muestra la geometría del modelo y se puede dividir hasta en cuatro ventanas para visualizar planos distintos del modelo.

3.2 Uso del Programa FEM

Para crear un modelo el usuario establece la grilla, posteriormente los nodos y elementos *beam* en relación a la grilla, asigna cargas, propiedades de sección y material a los elementos y asigna restricciones, masas concentradas y fuerzas a los nodos. El análisis de la estructura se realiza con base a los nodos, a los elementos *beam* y a sus propiedades. Los resultados del análisis matricial de la estructura se muestran gráficamente con los diagramas de fuerzas internas de los elementos *beam* y la deformación de la estructura debido a fuerzas estáticas. En el análisis modal se pueden ver las animaciones de los modos principales de la estructura y en una tabla sus periodos de vibración.

A continuación se proporciona los pasos básicos para realizar el modelo de una estructura.

1. Abrir un archivo nuevo
2. Establecer la grilla
3. Dibujar los nodos
4. Asignar restricciones a los nodos
5. Definir las propiedades de los elementos *beam*
6. Dibujar los elementos *beam*
7. Definir casos de carga
8. Asignar las cargas
9. Analizar el modelo

3.3 Análisis modal

El análisis modal del programa FEM ^[4] es lineal y utiliza vectores propios para determinar, por medio de la vibración libre, los modos y las frecuencias naturales del sistema. Estos modos naturales proveen una excelente interpretación del comportamiento de la estructura. El análisis de vectores propios involucra la solución generalizada del problema de valores propios:

$$(K - \Omega^2 * M) * \varphi = 0 \quad (1)$$

Donde K es la matriz de rigidez, M la matriz de diagonal de masa, Ω^2 es la matriz diagonal de los valores propios y φ es la matriz correspondiente de los vectores propios (formas modales).

Los vectores propios son llamados modos de vibración natural de la estructura. Estos modos son identificados por números del 1 hasta n en el orden en el cual los modos son encontrados por el programa.

3.4 Resultados

Para ejecutar el análisis del modelo se utiliza el menú Analyze - RunAnalyze, con esto el programa realizará los cálculos y se podrá acceder a la visualización de los resultados. Entre las opciones de visualización de resultados del programa encontramos la estructura deformada a cargas estáticas, animación de los modos de la estructura (Figura 3) entre otros datos de la estructura.

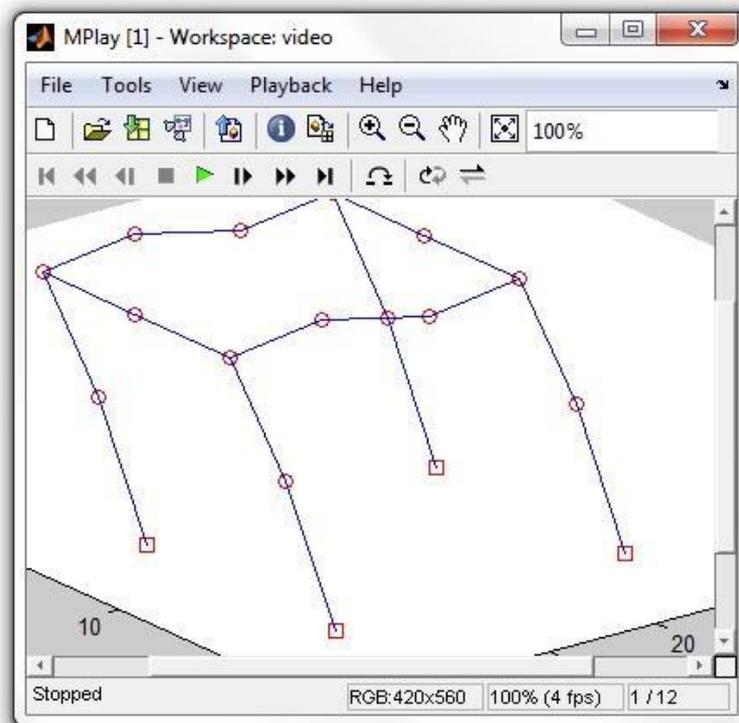


Figura 3. Animación de un modo de una estructura

Para los elementos podemos acceder a los diagramas de fuerzas internas haciendo clic derecho sobre el elemento en la opción *Diagram*. Se desplegará la ventana mostrada en la Figura 4, en la cual encontramos los diagramas de fuerzas internas con respecto a los ejes locales del elemento. Por último para los nodos podemos obtener las reacciones y desplazamientos haciendo clic derecho sobre el nodo.

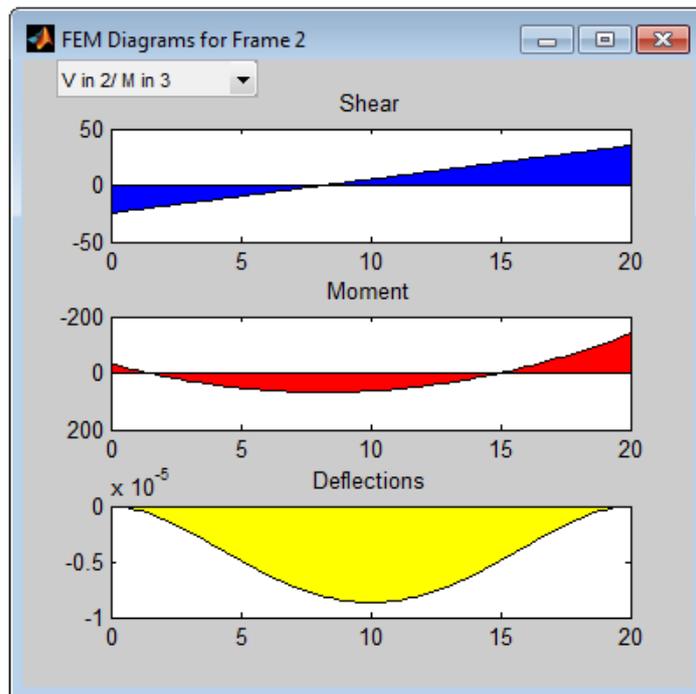


Figura 4. Diagrama de fuerzas internas de un elemento

4. CONCLUSIONES Y PERSPECTIVAS

Con el desarrollo de la interfaz gráfica se mejoró la funcionalidad y facilidad de uso respecto a la versión preexistente del programa FEM para la modelación de estructuras.

El lenguaje de programación de Matlab permitió el desarrollo de una interfaz amigable con el usuario permitiendo una entrada de datos sencilla para generar y modelar una estructura. Esto se logra utilizando a la programación orientada a objetos y los objetos de interfaz de usuario del programa Matlab.

Así, con el diseño de la interfaz gráfica, el *FEM* amplía su utilidad como herramienta de modelamiento, contribuyendo con su practicidad en el área académica, de investigación y en los diversos desarrollos profesionales.

El siguiente paso en el desarrollo de este programa, es incluir en la interfaz el modelado de elementos tipos shell y cables, la gráfica de las cargas aplicadas y la deformación de los elementos, entre otras mejoras para la segunda versión del programa.

AGRADECIMIENTOS

Los autores desean expresar su agradecimiento a Juan Martín Caicedo, creador de la caja de herramientas FEM y a Boris Zárate por sus aportes a la caja de herramientas FEM.

REFERENCIAS

- [1] Kong, X., Chen, D. An object-oriented design of FEM programs. Computers&Structures. Volume 57. 3 October 1995. Pages 157-166. ISSN 0045-7949
- [2] Flórez, R. Algoritmos, estructuras de datos y programación orientada a objetos. Esferaeditores, 2005. ISBN 958-648-394-0
- [3] Booch, G. Object-oriented analysis and design. Second. California: Addison Wesley. ISBN 0-8053-53402, 1998
- [4] Caicedo. J.M., Zarate, B., Marulanda, J., Thomson, P., Dyke, S.J. Desarrollo de una caja de herramientas para el análisis de estructuras por elementos finitos en Matlab. Artículo del II Congreso Colombiano de Ingeniería Sísmica. 2005
- [5] Zienkiewicz, O. El método de los elementos finitos. Reverte. ISBN 84-291-4894-9, 1981