CAPÍTULO 5: ALGUNAS ESTADÍSTICAS

A continuación se darán algunos datos del programa escrito para la realización de los análisis que aparecen en este proyecto:

- El código ocupa un total de 223,849 bytes, y dado que el formato de los archivos es tipo texto, dónde cada carácter ocupa un byte, puede afirmarse que el código tiene un máximo de 223,849 caracteres.
- Si cada carácter se colocase en una casilla de una libreta con una cuadrícula de 4x4 mm (libreta de cuadrícula fina típica), las páginas que estarían rellenadas de letras sobrarían para cubrir una cama de matrimonio típica.
- El programa consta de 26 funciones y 19 subrutinas de uso por el algoritmo.
- Las subrutinas no directamente utilizadas por el algoritmo suponen del orden de un 40% del código total.
- Si el código completo se escribiera en Word, sin espacios, letra tras letra, con un tamaño de letra de 12, podría llenar más de 90 páginas.
- El algoritmo que invierte las matrices es altamente eficiente, sin llegar al nivel computacional óptimo, logra tasas de realización de hasta 137,000 operaciones/segundo o 137 kFLOPS.
- Una matriz típica de estudio tiene 117 filas y columnas, lo que da un total esperable de $\frac{1}{2} \cdot 117^3$, es decir, un total de 800,807 operaciones necesarias.
- Si el ancho de la matriz el máximo posible, el algoritmo tardaría menos de seis segundos en realizar las 800,807 operaciones necesarias.

4.2.16. Esfuerzos locales en vigas

Con los desplazamientos locales de los nodos de cada viga, la subrutina EsfLocVigas calculará mediante la matriz de rigidez de cada viga los esfuerzos en cada una de ellas.

4.3. Resultados

4.3.1. Guardado del resumen

Al finalizar cada cálculo se llama a la subrutina GuardarDatosMemoria, de forma que el resumen de los datos encontrados se guarde en una parte semifija de la memoria RAM, separados del resto de resultados extraídos del cálculo actual.

4.3.2. Borrado del último puente

Para iniciar otro cálculo, mediante una llamada a la subrutina BorrarUltimoPuente se borra la memoria RAM que no contiene los resúmenes, pero liberando los recursos para realizar nuevos cálculos.

4.3.3. Guardado del contenido de la memoria

Cuando se han terminado de realizar simulaciones y cálculos, se vuelca el contenido de la memoria semifija a un archivo en el disco duro, mediante el la llamada a la subrutina VolcadoMemoria.

4.2.8. Generación de la matriz de rigidez acortada

Se llama a la función SubMatDespl, dando como parámetro la matriz global de rigidez, de forma que guardando el resultado tenemos la matriz global de rigidez sin los nodos empotrados al exterior.

4.2.9. Realización de la inversa de la matriz acortada

Se llama a la función Inversa y se guarda el resultado obtenido.

4.2.10. Generación del vector de fuerzas acortado

Se llama a la función SubVectDespl, dando como parámetro el vector de fuerzas global, de forma que guardando el resultado tenemos el vector de fuerzas global, pero sin los nodos empotrados al exterior.

4.2.11. Multiplicación de la inversa por el vector de fuerzas

Se llama a la función MultMatriz pasando estas dos matrices como parámetros, obteniéndose el vector de desplazamientos nodales en coordenadas globales.

4.2.12. Ampliado del vector de desplazamientos

Se amplia el vector de desplazamientos con los ceros correspondientes a los nodos empotrados al exterior.

4.2.13. Encontrar el vector de fuerzas global

Se multiplica con la función MultMatriz la matriz global de rigidez completa por el vector de desplazamientos ya ampliado, obteniéndose el vector de fuerzas global.

4.2.14. Encontrar las reacciones

Para hallar las reacciones se restan mediante la función RestMatrices, el vector de fuerzas global recién encontrado con el vector de fuerzas acortado que teníamos anteriormente, pero habiendo ampliado éste último antes de la resta, claro.

Ahora, en los nodos correspondientes a los empotramientos con el exterior aparecen las reacciones y los momentos.

4.2.15. Desplazamientos locales en vigas

Ahora, conocidos los desplazamientos de todos los nodos, se llama a la subrutina DesplLocVigas, que encontrará los desplazamientos locales de los nodos de cada viga.

4.1.3. Configuración de la simulación

De ser necesario un estudio que tenga en cuenta variaciones en los parámetros de entrada aquí se definirán tanto los valores mínimos, como máximos, como el valor de paso para cada parámetro.

Este punto correspondería al botón "Configurar simulación".

4.2. Cálculo

Ya se calcule un caso único o una serie de casos, el algoritmo utilizado es siempre el siguiente:

4.2.1. Creación de los puntos del puente

Mediante una llamada a la subrutina CalcularPuntosPuente, se genera en memoria un listado con las coordenadas de los puntos implicados.

4.2.2. Dibujo del puente en pantalla

Se realiza una llamada a la subrutina DibujarPuente para graficar la geometría en pantalla.

4.2.3. Creación de las barras

Se crea un listado en memoria de todas las barras que haya en el puente mediante la llamada a la función CrearListaBarra, definiéndose entonces sus propiedades y sus respectivas matrices de rigidez y fuerzas nodales debidas al peso propio.

4.2.4. Generación de cargas de uso

Se llama a la subrutina CrearCargasTablero, para definir los esfuerzos sobre este elemento.

4.2.5. Colocación de todas las barras en ejes globales

Se llama a la subrutina GirarBarras, para que todas terminen en ejes globales.

4.2.6. Montado de la matriz global de rigidez

A partir de las matrices de las barras en ejes globales y mediante una llamada a la subrutina MontarMatrizGlobal, se genera finalmente la matriz global de rigidez.

4.2.7. Montado del vector de fuerzas global

Se llama a la subrutina MontarVectorFGlobal para que monte el vector de fuerzas global a partir de los vectores de fuerzas en ejes globales de cada barra.

CAPÍTULO 4: DESCRIPCIÓN COMPLETA DEL ALGORITMO

4.1. Parámetros de entrada

4.1.1. Generación de geometría

En este paso se definen los parámetros básicos de geometría para un cálculo único; más adelante se definirán los rangos de variación, si se requieren.

Este punto correspondería al botón "Configurar geometría".

4.1.2. Definición de materiales y secciones

Ahora se definirán los parámetros que definen la rigidez de cada barra, tales como: módulo de Young, sección y módulo de inercia.

La longitud no es un parámetro que pueda editarse, puesto que viene impuesta por la geometría generada.

Este punto correspondería al botón "Configurar materiales".

hubiera anteriormente con el nuevo, , realizando las llamadas que sean necesarias a la subrutina SumarNodoMatrizGlobal.

3.3.8. MontarVectorFGlobal

Esta subrutina guarda en memoria un vector global de fuerzas nodales al que va añadiendo las submatrices 3x1 de cada barra en ejes globales, en la fila que corresponda, sumando automáticamente el número que hubiera anteriormente con el nuevo.

3.3.9. SumarNodoMatrizGlobal

Suma una submatriz 3x3 proveniente de una barra en la fila y columna indicada de la matriz global de rigidez.

3.3.10. SumarNodoVectorFGlobal

Suma una submatriz 3x1 proveniente de los esfuerzos aplicados en un nudo de una barra en la fila indicada del vector de fuerzas global.

3.2.2. RellenarLista

Rellena la lista de barras con sus propiedades para que puedan ser editadas o cambiadas. También genera, aunque no muestra en pantalla, las matrices de todas ellas.

3.3. Formulario "frmPrincipal"

3.3.1. CalcularPuntosPuente

Esta subrutina calcula los puntos sobre el plano que definirán el puente, a partir de los parámetros de entrada aportados por el usuario. No los dibuja, solo crea una lista de puntos en memoria.

3.3.2. CargasTablero

Coloca una serie de cargas puntuales sobre las barras que correspondan, pero siempre manteniendo una separación entre ellas tal como se indique.

Automáticamente realiza las llamadas que sean necesarias a la subrutina AddCargaPunt.

3.3.3. CrearListaBarras

Genera un listado de barras, a partir de los puntos generados por la subrutina CalcularPuntosPuente.

3.3.4. DibujarPuente

Es una subrutina meramente gráfica, que simplemente dibuja y une con líneas los puntos que ya han sido definidos en memoria por la subrutina CalcularPuntosPuente.

3.3.5. EscribirVectorFGlobal

Esta subrutina realiza un volcado del vector de fuerzas global generado por el programa y lo guarda en formato texto.

Es una reminiscencia de una versión antigua del programa, ya no tiene uso alguno.

3.3.6. GirarBarras

Lee la lista de barras que hay en memoria y, si tienen definido un ángulo de inclinación, realiza el cambio de base de acorde a este ángulo, utilizando, para ello las llamadas necesarias a la función GiraMatriz.

3.3.7. MontarMatrizGlobal

Esta subrutina guarda en memoria una matriz global de rigidez a la que va añadiendo las submatrices 3x3 de cada barra en ejes globales, en la fila y columna que corresponda, sumando automáticamente el numero que

3.1.2. BorrarTodo

Hace un borrado general de la memoria RAM, de forma que libera todos los recursos utilizados del sistema, aunque pierde los datos allí contenidos si no han sido previamente guardados al disco duro.

3.1.3. BorrarUltimoPuente

Borra de la memoria los datos del último puente que haya sido calculado, manteniendo la memoria semifija intacta, lo que permite conservar los resúmenes de los anteriores puentes calculados, y sin embargo, liberar la memoria para el cálculo del siguiente.

3.1.4. DesplLocVigas

Extrae los desplazamientos del vector de desplazamientos globales y los sitúa en el vector de desplazamientos global de cada viga, para posteriormente cambiarlos de base y guardarlos en el vector de desplazamientos local de cada viga.

3.1.5. EsfLocVigas

Calcula los esfuerzos locales que se dan en cada viga a partir de su matriz de rigidez en ejes locales y su vector de desplazamientos en ejes locales.

3.1.6. GuardarDatosMemoria

Guarda los datos generados para el último puente calculado en la memoria semifija, que contendrá los datos resumidos de todos los puentes calculados.

3.1.7. VolcadoMemoria

Realiza un guardado global de todos los datos almacenados en la memoria, de forma que queden guardados en un archivo Excel formato 2003, en el directorio raíz "C:\", cuyo nombre es la fecha y hora del momento del volcado de datos.

3.2. Formulario "frmMat"

3.2.1. CrearMatricesTemporales

Crea una lista con las matrices temporales de todas las barras, en ejes locales, para que las propiedades de las barras puedan ser modificadas, y cuando se confirmen los valores, puedan convertirse en las matrices definitivas en ejes locales.

También genera los vectores de fuerzas nodales para cada barra en ejes globales, a partir de sus densidades y medidas.

CAPÍTULO 3: DESCRIPCIÓN DE LAS SUBRUTINAS

Las subrutinas son fragmentos de código que debido al algoritmo, deben ser ejecutadas más de una vez y por ello se agrupan en estructuras que se puedan ejecutar para realizar una determinada acción, que aunque puede hacer cambiar el estado de las variables al llamar a otras funciones, no produce ningún resultado directo por sí misma.

Aquí se mostrarán las 19 subrutinas necesarias para el desarrollo normal del algoritmo y se omitirán las referencias a las subrutinas cuyo uso no sea totalmente necesario para la ejecución del algoritmo, tales como subrutinas de cambio de unidades.

Se ordenarán las funciones por módulo y por orden alfabético.

3.1. Módulo "General"

3.1.1. AddCargaPunt

Añade una carga puntual en el sentido vertical descendente, sobre el punto indicado de la barra indicada. Para ello accede al vector de fuerzas nodales de la barra y suma las nuevas fuerzas y momentos generados.

Se usa para situar las cargas de uso sobre el tablero.

Utiliza un algoritmo análogo al de resolución por el método de Gauss.

2.2.5. DiagonalizarSup

Convierte la matriz indicada en una matriz diagonal superior, pero cuyos pivotes no tienen porqué ser unos.

Utiliza un algoritmo análogo al de resolución por el método de Gauss.

2.2.6. GenMatAmpl

Genera una matriz a partir de la indicada, pero con el doble de filas, rellenando ese espacio vacío con la matriz identidad.

Se usa para poder utilizar más tarde el algoritmo de eliminación de Gauss-Jordan, de forma que la matriz identidad se haya convertido en la matriz inversa.

2.2.7. HacerCeros

Busca todos los valores menores al valor de tolerancia indicado y los sustituye por un cero.

Se usa para evitar que los ceros numéricos afecten al cálculo produciendo resultados tendientes a cero pero sin serlo.

2.2.8. Inversa

Amplia la matriz indicada, luego llama a las funciones DiagonalizarInf y DiagonalizarSup para que la matriz ampliada se haga igual a la identidad y finalmente descarta la matriz identidad y toma la matriz inversa que ha calculado.

2.1.16. SubVectDespl

Elimina los nodos que están empotrados al exterior y reduce el vector que se le da por parámetro.

Se utiliza para eliminar las filas correspondientes a los nodos de los apoyos, en el vector global de desplazamientos.

2.1.17. TrasponerMatriz

Transpone la matriz indicada y devuelve el resultado.

Se usa para poder realizar los cambios de base.

2.1.18. VectTexto

Convierte el vector estipulado en una cadena de texto.

Es una función auxiliar que se usa para poder mostrar el vector de una forma entendible por el usuario.

2.2. Módulo "cMatrices"

2.2.1. CambiaFilasInf

Busca (en orden descendente desde el pivote) una fila cuyo valor, en la columna indicada como parámetro, sea diferente de cero, y lo coloca en la fila de inicio indicada como parámetro.

Se utiliza para encontrar pivotes en la matriz que no sean cero. Si todos los valores de esa columna a partir del pivote para abajo son cero, no puede hacerse la inversa de la matriz.

2.2.2. CambiaFilasSup

Busca (en orden ascendente desde el pivote) una fila cuyo valor, en la columna indicada como parámetro, sea diferente de cero, y lo coloca en la fila de inicio indicada como parámetro.

Se utiliza para encontrar pivotes en la matriz que no sean cero. Si todos los valores de esa columna a partir del pivote para arriba son cero, no puede hacerse la inversa de la matriz.

2.2.3. ConvMatToText

Es una función que convierte una matriz en memoria en texto entendible por el usuario. Es una reminiscencia de una función usada por otro programa que llamaba a este módulo de código.

2.2.4. DiagonalizarInf

Convierte la matriz indicada en una matriz diagonal inferior, pero cuyos pivotes no tienen porqué ser unos.

2.1.9. Longitud

Devuelve la distancia entre dos puntos sobre el plano, a partir de sus coordenadas X e Y.

Se utiliza para calcular la longitud de las barras en función de las coordenadas de sus nodos inicial y final.

2.1.10. MatrizNodo

Devuelve la matriz de 3x3 correspondiente al nodo cuya fila y columna se estipulan, copiándolo la matriz de entrada.

Se utiliza para montar la matriz global de rigidez, haciendo que, de las matrices de cada barra se copie el contenido de cada nodo y se sume a la posición correspondiente de la matriz global de rigidez.

2.1.11. MatTexto

Convierte la matriz estipulada en una cadena de texto.

Es una función auxiliar que se usa para poder mostrar la matriz de una forma entendible por el usuario.

2.1.12. MultMatrices

Multiplica las dos matrices que le sean dadas como parámetros y devuelve el resultado, si éste puede calcularse.

Se utiliza para multiplicar la inversa de la matriz de rigidez por el vector de fuerzas global y para otros productos de matrices.

2.1.13. NomNudo

Devuelve el nombre del nudo que se indica mediante su número de índice de referencia interno en memoria.

Se utiliza como función auxiliar para conocer el nombre de los nodos en memoria. Es la función opuesta a IndiceNudo.

2.1.14. RestMatrices

Devuelve el resultado de la resta entre las matrices que se le den como parámetros, si es posible calcularlo.

Se utiliza para encontrar las reacciones a partir de la diferencia entre el vector de fuerzas globales y el producto del vector de desplazamientos y la matriz global de rigidez.

2.1.15. SubMatDespl

Elimina los nodos que están empotrados al exterior y reduce la matriz que se le da por parámetro.

Se utiliza para eliminar las filas y columnas correspondientes a los nodos de los apoyos, en la matriz global de rigidez.

2.1.2. Angulo

Devuelve el ángulo de la recta entre dos puntos sobre el plano bidimensional, medidos siempre desde el primer punto al segundo.

2.1.3. CreaMatriz

Crea la matriz de rigidez de una barra, en coordenadas locales de ésta, en función del tipo de barra, la creará como biempotrada o biarticulada.

Para crear la matriz se necesita que esta función reciba el módulo de Young del material, su sección, su momento de inercia y su longitud.

2.1.4. CreaMatrizGiro

Genera una matriz de giro a partir del ángulo indicado.

Se usa conjuntamente con la función TrasponerMatriz para poder hacer cambios de coordenadas de las matrices de las barras.

2.1.5. CreaVectorF

Crea el vector de fuerzas nodales en ejes globales que se aplicará sobre los nudos de los extremos de la barra.

Utiliza como parámetros la densidad del material, la sección y longitud de la barra y el ángulo de ésta.

Se utiliza para poder definir más tarde el vector de fuerzas nodales global.

2.1.6. GiraMatriz

Esta función gira una matriz el ángulo indicado y devuelve el resultado. Para ello automáticamente llama a la función CreaMatrizGiro y a la función TrasponerMatriz.

Se utiliza para el cambio de coordenadas en las barras.

2.1.7. GiraVector

Esta función gira un vector el ángulo indicado y devuelve el resultado. Para ello automáticamente llama a la función CreaMatrizGiro.

Se utiliza para el cambio de coordenadas en los vectores.

2.1.8. IndiceNodo

Devuelve el índice de referencia interno en memoria del nodo cuyo nombre coincida con el valor de entrada estipulado.

Se utiliza como función auxiliar para buscar nodos en memoria. Es la función opuesta a NomNudo.

CAPÍTULO 2: DESCRIPCIÓN DE LAS FUNCIONES

En este capítulo se describirán las 26 funciones que implementa el programa y que son totalmente necesarias de un modo u otro para poder seguir el algoritmo de cálculo, y se omitirán aquellas cuyo uso, aunque necesario no sea irrenunciable, tales como las funciones de conversión de unidades y similares.

Se indicarán cuales son sus parámetros de entrada y de salida y la acción que realizan.

Se ordenarán las funciones por módulo y por orden alfabético.

2.1. Módulo "General"

2.1.1. AmpliarVectorDGlobal

Esta función recibe un vector sin las filas correspondientes a los nodos de apoyo, y devuelve un vector cuyos valores son los mismos del vector anterior, solo que añadiendo los ceros pertinentes en las posiciones adecuadas, allí dónde estarían los nudos empotrados al exterior.

Se utiliza para ampliar el vector de desplazamientos y de fuerzas surgidos de la multiplicación del vector de fuerzas nodales por la inversa de la matriz de rigidez.

CAPÍTULO 1:

OBJETO Y ALCANCE

1.1. Objeto

El objeto de este documento es servir de apoyo para la realización de una aplicación que dado un intervalo de variación de cuatro parámetros de entrada, a saber: luz del tablero, altura de las torres, separación entre tirantes y ángulo de las torres; nos genere la geometría básica de un puente atirantado y posteriormente utilice el método matricial de la rigidez para calcular los desplazamientos, y esfuerzos generados en cada nudo, y por tanto en cada barra, presentando los resultados obtenidos en un formato de fácil manipulación como pueda ser Excel.

1.2. Alcance

En este documento no se tratarán los detalles internos de funcionamiento básico de los algoritmos del programa, si no que en su lugar se describirán las acciones que el programa realiza, pudiéndose consultar los archivos de código fuente adjuntos en el CD o DVD que acompaña este proyecto.

El programa trabajará con estructuras de tipo bidimensional, puesto que, por falta de tiempo no se ha creído necesaria la implementación del cálculo de estructuras tridimensionales.

Deberá poder calcular todos los desplazamientos de todos los nodos de la estructura y utilizar esta información a posteriori para obtener las fuerzas en los nudos.

A partir de estos esfuerzos, debe poder calcular los esfuerzos en cada barra.

ÍNDICE APLICACIÓN