Instituto Tecnológico de Costa Rica

Métodos numéricos

Profesor: Alfredo Rodríguez

Tarea programada #2

Estudiantes

Dávila Rodríguez Stephanny 2019033581

Rodríguez Morales Marco 2019163031

Rodríguez Rivas Daniel 2019039694

Descripción del trabajo

En el presente trabajo se desarrollará un ejercicio sobre una estructura mecánica, a la cual, mediante el método de Gauss-Seidel y Factorización LU, se le obtendrán las reacciones en dos de sus nodos.

Los programas utilizados corresponden a métodos numéricos que obtienen el resultado deseado después de cierta cantidad de iteraciones o cuando se tenga un error deseado introducido por el usuario.

Los resultados obtenidos, al ser producto de procesos recursivos, no son exactos al 100 por ciento, sin embargo, estos pueden aproximarse más al resultado verdadero en función del error al que se desee llegar y la cantidad máxima de iteraciones que el usuario defina.

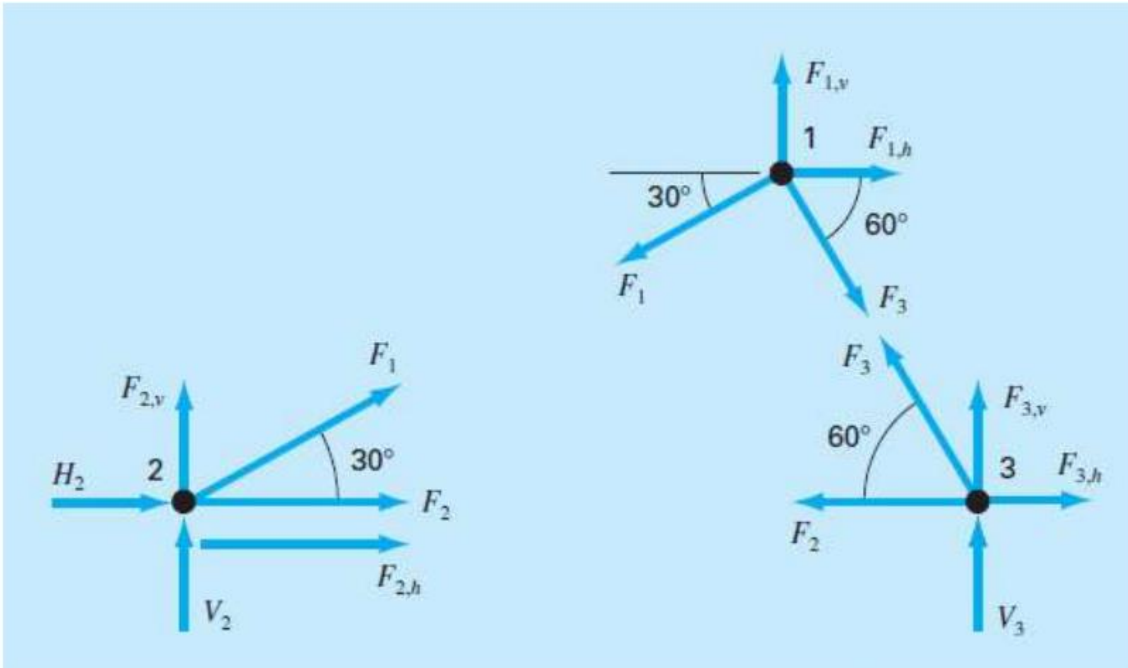

Figura 1 Diagrama de cuerpo libre del elemento estructural a estudiar. Fuente: [1]

#1 **6ptos**

Nodo 1:

En x:

$F1h+F3\cos(\alpha)-F1\cos(\beta)=0$

En y:

$F1v-F3\sin(\alpha)-F1\sin(\beta)=0$

Nodo 2:

En x:

$H2+F1\cos(\beta)+F2h+F2=0$

En y:

$V2+F2v+F1\text{sen}(\beta)=0$

Nodo 3:

En x:

F3h-F2-F3cos(α)=0

En y:

F3v+V3+F3sen(α)=0

#2 **3ptos**

Nodo 1:

En x: F1h + 0.573 - 0.819F1 = 0

En y: F1v - 0.819 - 0.574F1 = 0

Nodo 2:

En x: H2 + 0.819F1 + F2h + F2 = 0

En y: V2 + F2v + 0.574F1 = 0

Nodo 3:

En x: F3h - F2 - 0.573F3 = 0

En y: F3v + V3 + 0.819F3 = 0

$$A = \begin{vmatrix} 0.819 & 0 & -0.573 \\ 0 & 1 & 0.574 \\ 0.819 & 0 & 0.819 \end{vmatrix}$$

No es diagonal dominante, por lo tanto, no cumple con el criterio de convergencia para Gauss-Seidel y Factorización LU.

#3 **8ptos**
No se puede acomodar para que cumpla con el criterio de convergencia para el método Gauss-Seidel.

Verificación:

$$A = \begin{vmatrix} 0.819 & 0 & -0.573 \\ 0 & 1 & 0.574 \\ 0.819 & 0 & 0.819 \end{vmatrix}$$

Fila 1: |0.819|>|-0.573|+|0| -> 0.819 > 0.573 **CUMPLE**

Fila 2: |1 |>|0.574|+|0| -> 1 > 0.574 **CUMPLE**

Fila 3: |0.819|>|0.819|+|0| -> 0.819 > 0.819 **NO CUMPLE**

Al aplicar la verificación, se puede observar que la fila 1 y 2 cumplen, sin embargo, la fila 3 no cumple, siendo este el problema, debido a que el método de pivoteo visto en clase (intercambio de filas) no soluciona el problema.

#4 **7ptos**

Primero recibe como entrada una matriz A, seguidamente se verifica si la matriz es cuadrada, si no lo es, generará un mensaje de error. Luego inicializa las matrices L y U, L como matriz de identidad y U igual a la matriz de entrada A, y el vector de permutación como una secuencia de enteros. Luego utiliza un ciclo para iterar a través de cada columna de la matriz y realiza pivoteo parcial para intercambiar filas en U y, si corresponde, en L para llegar a tener la condición de diagonal dominante. Posteriormente, lleva a cabo la eliminación gaussiana para actualizar las matrices L y U en cada iteración. Finalmente, la función devuelve las matrices L y U resultantes.

#5 **5ptos**

La función primero recibe una matriz A y un vector b, seguidamente verifica si la matriz es cuadrada y si el tamaño del vector coincide con el número de filas de A. Luego inicializa un vector “sol” de ceros del tamaño del número de filas de A. Luego utiliza un bucle para iterar a través de cada fila de A. En cada iteración, calcula el valor de la solución correspondiente usando la fórmula de sustitución hacia adelante, que implica restar la suma del producto de los elementos conocidos de A y solución previamente calculada, y luego dividir por el elemento de A en la diagonal correspondiente. Finalmente, la función devuelve el vector sol que contiene la solución del sistema de ecuaciones lineales.

#6 **5ptos**

La función primero recibe una matriz A y un vector b, seguidamente verifica si la matriz es cuadrada y si el tamaño del vector coincide con el número de filas de A. Luego inicializa un vector sol de ceros del tamaño del número de filas de A. Utiliza un bucle que itera en reversa, comenzando desde la última fila de A y avanzando hacia la primera fila. En cada iteración, calcula el valor de la solución correspondiente utilizando la fórmula de sustitución hacia atrás, que implica restar la suma del producto de los elementos conocidos de A y la solución previamente calculada, y luego dividir por el elemento de A en la diagonal correspondiente. Al final, la función devuelve el vector “sol” que contiene la solución del sistema de ecuaciones lineales.

#7 **5ptos**

La función recibe el vector de constantes de un sistema y lo resuelve utilizando las funciones FactorizacionLU, SustAdelante y SustAtras. Estas funciones se llaman dentro de la misma función y ellas proporcionan los valores necesarios para poder llamar a la siguiente función e ir obteniendo el resultado final.

#8 **7ptos**

```
function [X, error, iter] = gaussseidel(A, b, Tol, iterMax)
    [m, n] = size(A);

    if m ~= n
        error('La matriz A debe ser cuadrada.');
    end

    if length(b) ~= m
        error('El tamaño del vector b debe coincidir con el número de filas de A.');
    end

    % Verificar si la matriz A es diagonal dominante
    if ~esDiagonalDominante(A)
        warning('La matriz A no es diagonalmente dominante, los resultados pueden no converger.');
    end

    % Inicializar la matriz X con ceros
    X = zeros(m, 1);

    % Inicializar el vector de errores
    error = zeros(iterMax, 1);

    % Realizar iteraciones
    for iter = 1:iterMax
        X_ant = X;  % Almacenar la solución anterior

        for i = 1:m
            suma = A(i, :) * X - A(i, i) * X(i);
            X(i) = (b(i) - suma) / A(i, i);
        end

        % Calcular el error en esta iteración utilizando norma 4
        error(iter) = norm(X - X_ant, 4);

        % Verificar la convergencia
        if error(iter) < Tol
            break;
        end
    end
```

```
    % Recortar el vector de errores al número de
iteraciones realizadas
    error = error(1:iter);

    % Si no se alcanza la tolerancia, mostrar un mensaje
de advertencia
    if iter == iterMax && error(iter) > Tol
        warning('El método de Gauss-Seidel no ha
convergido dentro del número máximo de iteraciones.');
    end
end

function esDominante = esDiagonalDominante(A)
    [m, n] = size(A);
    esDominante = true;

    for i = 1:m
        diagonal = abs(A(i, i));
        suma = sum(abs(A(i, :))) - diagonal;

        if diagonal <= suma
            esDominante = false;
            break;
        end
    end
end
```

#9

Parámetros de entrada para ambas funciones

```
F1_h = F1_mag * cosd(angulo);
F1_v = F1_mag * sind(angulo);

A = [0.819, 0, -0.574, 0, 0, 0;
     0, 1, 0.574, 0, 0, 0;
     0.574, 0, 0.819, 0, 0, 0;
     -0.819, -1, 0, -1, 0, 0;
     -0.574, 0, 0, 0, -1, 0;
     0, 0, -0.819, 0, 0, -1];

b = [F1_h; 0; F1_v; 0; 0; 0];
```

a) **Resolver el sistema utilizando Gauss-Seidel 4ptos**

```
[X_gaussseidel, ~, ~] = gaussseidel(A, b, 0.01, 1000);
```

```
Resolviendo el sistema utilizando Gauss-Seidel:
Reacciones y fuerzas calculadas con Gauss-Seidel:
F1 = -906.39 lb
F2 = 242.33 lb
F3 = -422.17 lb
H2 = 500.00 lb
V2 = 520.27 lb
V3 = 345.76 lb
```

b) **Resolver el sistema utilizando la factorización LU 5ptos**

```
[L, U] = factorizacionLU(A);
```

```
Resolviendo el sistema utilizando Factorización LU:
Reacciones y fuerzas calculadas con Factorización LU:
F1 = -906.38 lb
F2 = 242.33 lb
F3 = -422.17 lb
H2 = 500.00 lb
V2 = 520.26 lb
V3 = 345.76 lb
```

#10 **15ptos**

Sí se esperaba que el método Gauss-Seidel convergiera a la respuesta utilizada porque la matriz "A" utilizada era diagonal dominante.

#11 **10ptos**

Parámetros de entrada

```
A = [-1, 0, 0, 0, -0.707, 0, 0.707, 0, 0, 0;
     1, -0.866, 0, 0.5, 0, 0, 0, 0, 0, 0;
     0, 0.866, 1, 0, 0, 0, 0, 0, 0, 0;
     0, 0.5, 0, 0.866, 0, 0, 0, 0, 0, 0;
     0, 0, 0, -0.866, -0.707, 0, 0, 0, 0, 0;
     0, 0, -1, -0.5, 0.707, 1, 0, 0, 0, 0;
     0, 0, 0, 0, 0.707, 0, 0.707, 0, 0, 0;
     0, -0.5, 0, 0, 0, 0, 0, -1, 0, 0;
     0, 0, 0, 0, 0, -1, -0.707, 0, -1, 0;
     0, 0, 0, 0, 0, 0, -0.707, 0, 0, -1];


b = [0; 0; 0; -200; 0; 0; -400; 0; 0; 0];
```

Parámetros de entrada para Gauss-Seidel

```
[X_gaussseidel, ~, ~] = gaussseidel(A, b, 0.01, 1000);
```

Parámetros de entrada para Factorización LU

```
[L, U] = factorizacionLU(A);
```

El método Gauss-Seidel no converge ya que no hay forma de convertir la matriz a diagonal dominante por medio de los métodos vistos en clase. Los resultados se pueden ver en la siguiente imagen:

```
Warning: El método de Gauss-Seidel no ha convergido dentro del número máximo de iteraciones.
> In gaussseidel (line 41)
  In EJERCICIO (line 86)
Reacciones y fuerzas calculadas con Gauss-Seidel:
F1 = -815485021529058782020548104247185690083517275015365187010442845987354253330331085565727091947245
F2 = -121351290479232077314301641302106877445815768408763245032921342193131714391276023983620334048233
F3 = 105090217555014984437236408467145147418420261684279360695953245269712961254390890890492260729 1456
F4 = 700642554729977244348351471082869957342015543376393440663979332915657500892376640004389911543 5885
F5 = -858212803954965157264222171126839855189287452746408666423326228656433986164895942744996651801252
F6 = 200797990531129924853039325024654042304177525766305586717560211837500851606757259791250659700 9404
F7 = 858212803954965157264222171126839855189287452746408666423326228656433986164895942744996651801252 6
V3 = 606756452396160298702098156838730035460748613801983044115834869131349331707247842343148922751 1675
H5 = -261473635770745963510190145675707481027085409970687209234020882934066708802395271783060826725056
V5 = -606756452396160298702098156838730035460748613801983044115834869131349331707247842343148922751167

Resolviendo el sistema utilizando Factorización LU:
Reacciones y fuerzas calculadas con Factorización LU:
F1 = -375.12 lb
F2 = -424.88 lb
F3 = 367.94 lb
F4 = 14.36 lb
F5 = -17.59 lb
F6 = 387.56 lb
F7 = -548.18 lb
V3 = 212.44 lb
H5 = -0.00 lb
V5 = 387.56 lb
```

Bibliografía

[1] Chapra, S. C., & Canale, R. P. (2015). Métodos numéricos para ingenieros Quinta Edicion. Mexica: McGraw-Hill Interamericana.