
95.10 | Modelación numérica

75.12 | Análisis numérico I A

95.13 | Métodos matemáticos y numéricos

Trabajo Práctico 1 – Cuatrimestre 1 2023
Introducción a la programación

Paz Blanco, Pilar	105600
Méndez San Antonio Sara Angélica	107418
Covini, Juan Pablo	101114

Fecha	Correcciones / Observaciones	Docente

Calificación Final	Docente	Fecha

Índice

1. Introducción.....	3
2. Metodología.....	3
3. Resolución	4
Armar el sistema de ecuaciones lineales a resolver para $n=10$, $n=50$ y $n=100$	4
Resolver el sistema para las tres discretizaciones propuestas	4
Análisis de sensibilidad.....	6
Número de Condición del Problema	¡Error! Marcador no definido.
4. Conclusiones	9
5. Anexo:	11
Códigos	11
Resultados de Deformación de la Viga.....	15

1. Introducción

En el siguiente informe se busca calcular y analizar la curva elástica de una viga simplemente apoyada, partir de la utilización de distintos métodos (tanto directos como indirectos) de resolución. A su vez, se busca comparar y analizar los posibles motivos de errores los cuales se evidencian en las diferencias de los resultados obtenidos al utilizar los distintos métodos.

2. Metodología

Para el siguiente informe se utilizarán los métodos de Jacobi, Gauss-Seidel y eliminación Gaussiana para calcular la deformación de la viga, utilizando la siguiente ecuación:

$$q(x) = E \cdot I \cdot \frac{d^4 v}{dx^4}$$

Donde:

- $q(x)$: carga aplicada sobre la viga.
- E : módulo de Young, utilizando valor $E = 1$.
- I : momento de inercia, utilizando valor $I = 1$.

Para resolver esta ecuación vamos a realizar una aplicación de diferencias finitas, discretizando el dominio, tomando como condiciones de borde que:

$$\begin{aligned} v(0) &= 0 & \frac{d^4 v}{dx^4}(x=0) &= 0 \\ v(L) &= 0 & \frac{d^4 v}{dx^4}(x=L) &= 0 \end{aligned}$$

De esta forma construimos un sistema (**Sistema De Ecuaciones 1**) donde:

$$\begin{aligned} i=0 & & u_0 &= 0 \\ i=1 & & -4u_0 + 5u_1 - 4u_2 + u_3 &= f_1 \\ 1 < i < n-1 & & u_{i-2} - 4u_{i-1} + 6u_i - 4u_{i+1} + u_{i+2} &= f_i \\ i=n-1 & & u_{n-3} - 4u_{n-2} + 5u_{n-1} - 4u_n &= f_{n-1} \\ i=1 & & u_n &= 0 \end{aligned}$$

Donde

$$f_i = \frac{q(x_i)}{E \cdot I} \cdot \left(\frac{L}{n}\right)^4 \quad \text{(Ecuación 1)}$$

$$x_i = i \cdot L/n$$

$$q(x_i) = 2 + 4 \cdot (x - x^2) \quad \text{(Ecuación 2)}$$

3. Resolución

Todos los códigos y tablas referenciados se encuentran en el anexo.

Armar el sistema de ecuaciones lineales a resolver para $n=10$, $n=50$ y $n=100$

Utilizando el sistema de ecuaciones y las fórmulas planteadas en metodología, armamos dos matrices:

1. Matriz “A”: corresponde al lado izquierdo del **Sistema de Ecuaciones 1**. Se realiza con el **Código 1**.
2. Matriz “B”: corresponde al lado derecho del Sistema de ecuaciones, es decir que si lo pensamos como $A \cdot x = b$, se corresponde con “b”. Se realiza con el **Código 2**.

Ambas matrices se pueden encontrar resueltas en el Anexo, para 10, 50 y 100 nodos.

Resolver el sistema para las tres discretizaciones propuestas

Los métodos de resolución planteados son los siguientes:

1. Eliminacion Gaussiana
2. Jacobi
3. Gauss – Seidel

El código utilizado **para los mismos y los resultados se encuentran en el anexo como Código 3, 4 y 5 respectivamente**. A continuación, realizaremos un análisis de cada método, indicando los errores y la complejidad temporal, entre otros.

Se observa que los resultados obtenidos para Eliminación Gaussiana y Gauss-Seidel corresponden a los esperados para el modelo físico estudiado. Esto puede apreciarse mejor al graficar los resultados, donde se puede ver que los mismos tienen una forma similar a una parábola, lo que podríamos esperar de una deformación de la viga al aplicarle una fuerza. También se observa que el método de Jacobi no converge para ninguna cantidad de nodos propuesta.

Por otra parte, al analizar los resultados obtenidos por el método de Eliminación Gaussiana y los obtenidos por el método de Gauss-Seidel, se observan ligeras diferencias en las magnitudes de los resultados de los mismos, en especial para cantidades bajas de nodos.

Estas dos observaciones llevan a pensar que puede existir un problema con la condición del problema y sus errores. Al analizar el radio espectral de la matriz de iteración T se observa lo siguiente:

Radio Espectral (T)	Jacobi	Gauss-Seidel
N=10	1.56	4.95
N=50	1.66	5.23
N=100	1.75	5.55

Tabla 1: radio espectral de cada método en base a los nodos.

Se observa que, en ninguno de los casos planteados, el radio espectral de T es menor o igual a 1, lo que no garantiza la convergencia de los métodos en ninguno de los casos estudiados.

Este análisis indica que es necesario estudiar el número de condición del problema $K(A)$, el cual nos indicará si el problema se encuentra bien o mal condicionado:

	$K(A)$
N=10	1723
N=50	$1,0259 \cdot 10^6$
N=100	$1,6422 \cdot 10^7$

Tabla 2: $K(A)$ para las distintas cantidades de nodos.

Esto indica que el problema posee un factor de amplificación de errores muy grande, por lo que podemos ver que se encuentra muy mal condicionado. Esto explica las diferencias entre los resultados propuestos por la Eliminación Gaussiana y Gauss-Seidel para $n=10$.

Es interesante remarcar que, incluso con una gran amplificación de los errores, para cantidad de nodos altos, los resultados propuestos por Gauss-Seidel y Jacobi son aproximadamente iguales.

Analizando el orden de convergencia del método de Gauss-Seidel se obtienen los siguientes resultados:

	ρ
N=10	0.85
N=50	2.00
N=100	2.53

Tabla 3: orden de convergencia de Gauss-Seidel para distinta cantidad de nodos.

Se observa que el orden de convergencia para $n=10$ es menor a 1, lo que explica el alto número de iteraciones para una cantidad tan baja de nodos, cuando se espera que conlleve una baja cantidad de operaciones para resolver ese número de incógnitas. Cabe resaltar que el número de iteraciones para 10 nodos resulta similar a los números de iteraciones resultantes para $n=50$ y $n=100$. Esto indica que la potencia del método no se aprecia para valores bajos, lo que se refleja en un resultado diferente al obtenido por Eliminación Gaussiana y en una alta cantidad de iteraciones, situaciones que no ocurren para cantidades de nodos superiores.

	Gauss - Seidel	Jacobi
10	449	1624
50	990	1441
100	996	1442

Tabla 4: Cantidad de iteraciones realizadas en los métodos indirectos.

Una vez realizados estos análisis, se procedió a realizar perturbaciones en la solución inicial y la tolerancia con el fin de evaluar su impacto en la convergencia de los métodos indirectos empleados y los resultados obtenidos.

Se propusieron variaciones en la tolerancia, con el objetivo de lograr converger el método de Jacobi. Para ninguna de las cantidades de nodos planteados, un aumento o disminución de la tolerancia logra hacer converger este método.

Siguiendo dicho análisis, una disminución de la tolerancia para el método de Gauss-Seidel tiene efectos diferentes según el número de nodos que se trate. Para el caso $n=10$, se observa que el resultado obtenido es más parecido al obtenido por el método de Eliminación Gaussiana y el obtenido por Gauss-Seidel para mayor número de nodos. Por lo que se gana en la estimación realizada por el modelo, aunque se aumenta de manera significativa el costo computacional dado que se quintuplica el número de iteraciones. Esto se aprecia también en el orden de convergencia obtenido, el cual es menor al obtenido para la tolerancia propuesta por el resultado, con tolerancia 0,001 obtenemos 0,40 con $n=10$, 1,00 con $n=50$ y 1,35 con $n=100$.

Sin embargo, para las cantidades de nodos $n=50$ y $n=100$, un aumento o disminución de la tolerancia conlleva a que el modelo no converja.

Esta disparidad en los resultados obtenidos para la nueva tolerancia según el número de nodos muestra nuevamente los problemas de condicionamiento del problema, los cuales limitan la capacidad del modelo para alcanzar tolerancias más exigentes. Como consecuencia, el modelo muestra ser estable sólo para ciertos rangos de parámetros.

Por otra parte, se analizó reemplazar la solución con la que se comienza la iteración (x_0) por la solución obtenida por Eliminación Gaussiana para el número de nodos correspondientes. En este caso, para cualquier número de nodos y tolerancia, tanto el método de Jacobi como el de Gauss-Seidel convergen en una sola iteración al resultado, el cual es el mismo al x_0 planteado.

Esta sensibilidad a la solución inicial muestra nuevamente las dificultades presentadas por el problema.

Análisis de sensibilidad.

El siguiente análisis se realizará con 10 nodos para poder visualizar más claramente la variación en los resultados.

Variación del Módulo de Young: Gauss – Seidel:

E = 1	E = 2	Diferencia	Porcentual
0	0	0	
0.0026936	0.0013468	0.001347	50
0.005103	0.0025515	0.002552	50
0.0070383	0.0035192	0.003519	50.0007104
0.0083645	0.0041823	0.004182	50.0005978
0.0089749	0.0044875	0.004487	50.0005571
0.0087826	0.0043913	0.004391	50
0.0077306	0.0038653	0.003865	50
0.005821	0.0029105	0.002911	50
0.0031579	0.0015789	0.001579	49.9984167
0	0	0	

Tabla 5: Resultados obtenidos para 10 nodos($E=2$).

Se puede ver que, al duplicar el Módulo de Young, los resultados son la mitad que con el valor predeterminado de $E = 1$, por lo que podemos intuir que es inversamente proporcional.

Esto se ve reforzado al calcular los resultados con $E = 0,5$ (la mitad del valor predeterminado), donde podemos ver que el resultado se duplica. (**Tabla 6**)

$E = 1$	$E = 0.5$	Diferencia	Porcentual
0	0	0	
0.0026936	0.0053871	-0.00269	199.996287
0.005103	0.010206	-0.0051	200
0.0070383	0.014077	-0.00704	200.005683
0.0083645	0.016729	-0.00836	200
0.0089749	0.01795	-0.00898	200.002228
0.0087826	0.017565	-0.00878	199.997723
0.0077306	0.015461	-0.00773	199.997413
0.005821	0.011642	-0.00582	200
0.0031579	0.0063157	-0.00316	199.996833
0	0	0	

Tabla 6: Resultados obtenidos para 10 nodos($E=0.5$).

Las tablas anteriores se pueden ver plasmadas en el siguiente gráfico:

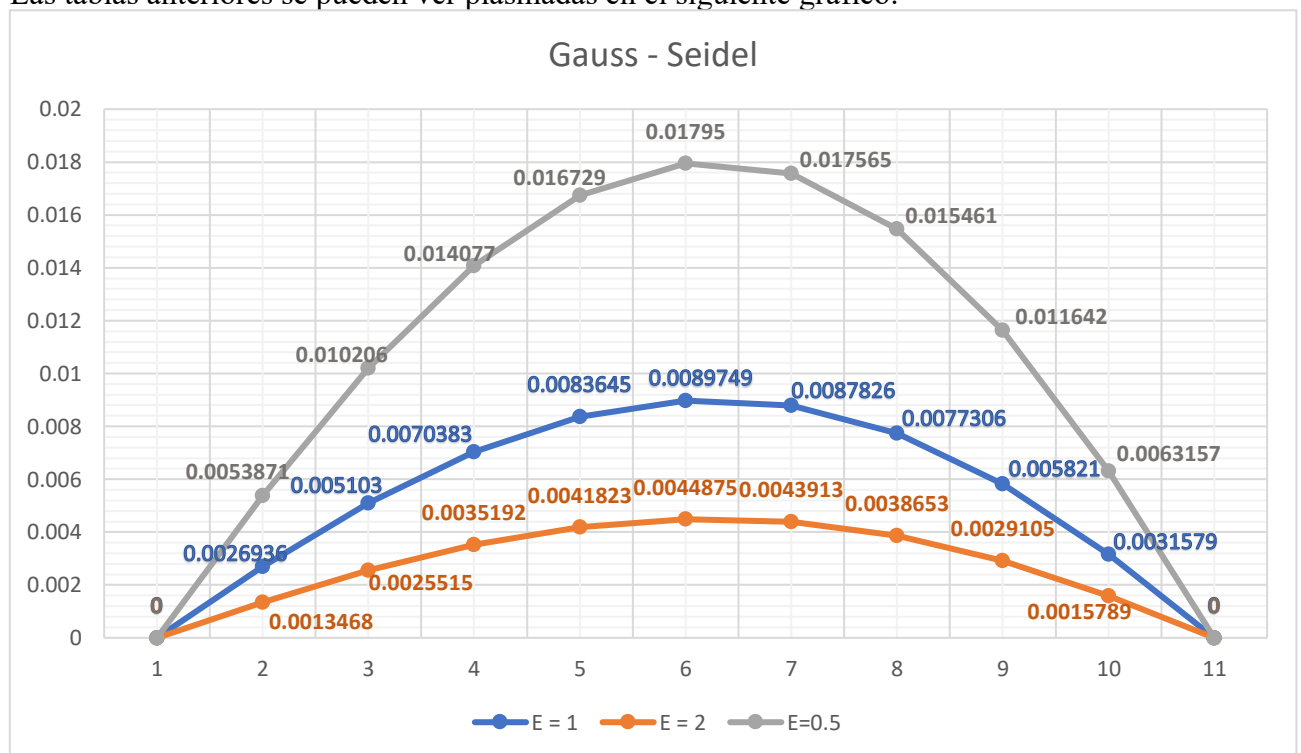


Gráfico 1: Resultados obtenidos en Gauss Seidel variando el Modulo de Young.

Variación del Módulo de Young: Gauss:

E = 1	E = 2	Porcentual E = 2	E = 0.5	Porcentual - E = 0.5
-1.91E-17	-9.54E-18	50.00	-3.82E-17	200.00
0.011583	0.0057915	50.00	0.023166	200.00
0.021936	0.010968	50.00	0.043872	200.00
0.030065	0.015032	50.00	0.06013	200.00
0.03524	0.01762	50.00	0.07048	200.00
0.037015	0.018507	50.00	0.07403	200.00
0.03524	0.01762	50.00	0.07048	200.00
0.030065	0.015032	50.00	0.06013	200.00
0.021936	0.010968	50.00	0.043872	200.00
0.011583	0.0057915	50.00	0.023166	200.00
0	0		0	

Tabla 7: Resultados obtenidos con Gauss al variar el módulo de Young.

Al igual que con Gauss-Seidel, si bien no obtenemos los mismos valores, el porcentaje en el que afecta el resultado la variación del módulo de Young podemos ver que es el mismo.

Esto también se ve de forma gráfica, donde si bien son distintos valores, la función tiene la misma forma.

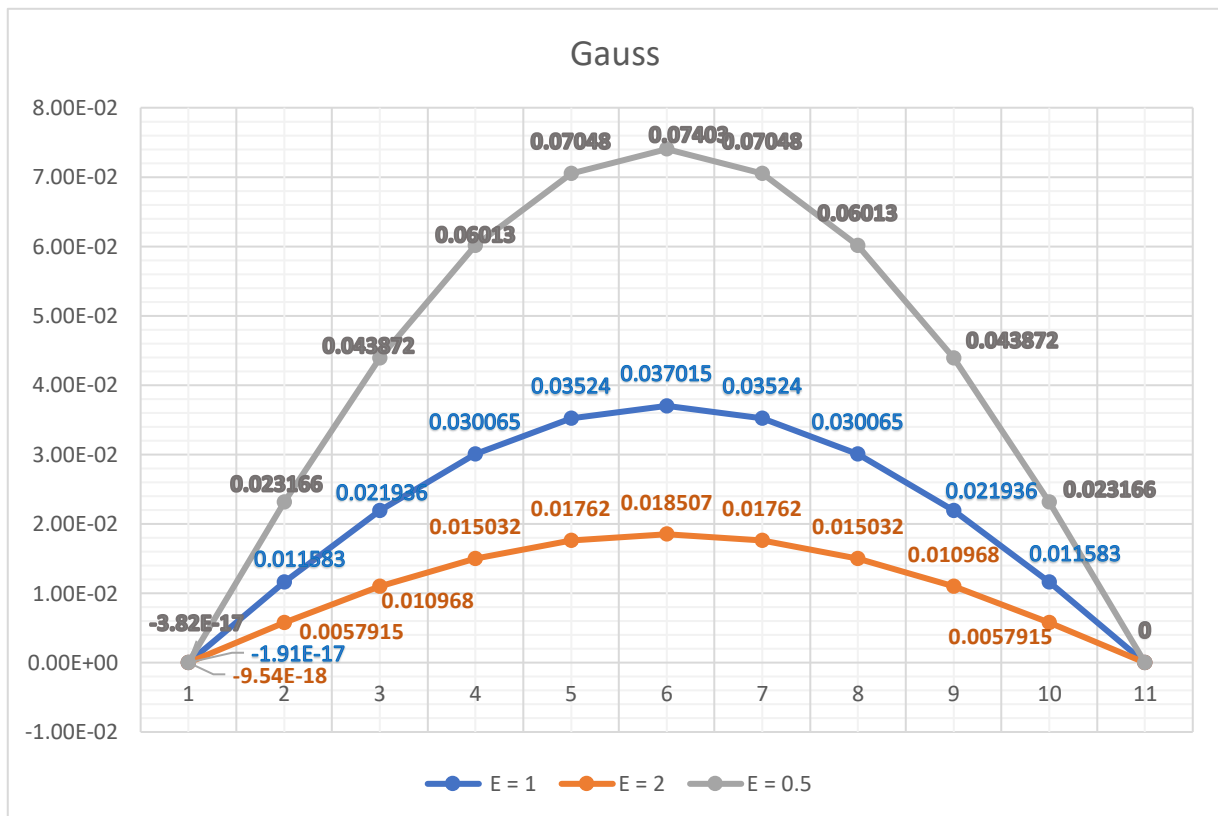


Gráfico 2: Resultados obtenidos en Gauss variando el Modulo de Young

4. Conclusiones

El problema estudiado presenta una gran amplificación de los errores, lo cual se expresa en número de condiciones muy altos para cada una de las cantidades de nodos estudiadas.

Este mal condicionamiento limita las capacidades de los métodos indirectos propuestos como solución, ya sea que los mismos no logran converger al resultado establecido o bien que si convergen logran hacerlo en rangos de parámetros muy limitados (como el caso de la modificación de la tolerancia para Gauss-Seidel).

Por otra parte, se observa que el problema se ajusta mejor a la resolución por métodos indirectos a mayores cantidades de nodos propuestos. Esto puede deberse a que requiera mayor número de iteraciones para “purgar” los errores propuestos por el problema.

El orden de convergencia varía mucho según el número de nodos y tolerancia propuestos, obteniendo así un número de iteraciones diverso según la cantidad de nodos propuestos.

A su vez al tratarse de un problema que se resuelve discretizando la solución, al utilizar cada vez más nodos es una representación más fiel a la realidad y sumado a que podemos ver que surgen muchos errores y situaciones no optimas con 10 nodos, es acorde a lo visto que con 50 nodos y 100 nodos los resultados sean más acordes a lo esperado. Sin embargo, discretizar de más el problema lleva a mayores costos computacionales, cuando entre 50 y 100 nodos, las soluciones se encuentran casi con la misma cantidad de iteraciones (en los métodos indirectos esa diferencia es menor a 6 iteraciones).

Se deberán explorar otras modelizaciones del problema posibles, ya sea incorporando nuevas variables, revisando la metodología empleada, entre otras opciones.

Al interpretar los datos que obtuvimos se puede ver que, para resultados del mismo método, pero variando la cantidad de nodos, nos da mayor precisión a lo largo de la curva, pero que este factor no afecta como tal los resultados (Gráfico 3), en cambio al variar el módulo de Young si es un factor que los afecta (Gráfico 2).

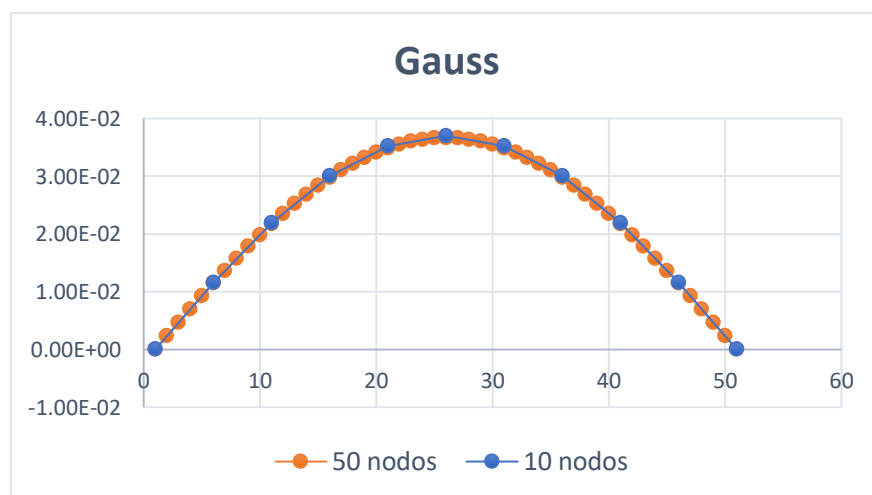


Gráfico 3: Resultados Gauss, 10 y 50 nodos

En cambio, cuando comparamos los resultados entre los distintos métodos podemos apreciar que la curva que describen ambos son similares, pero con órdenes de magnitud diferentes, esto se ve entre el método de Gauss, y el método de Gauss-Seidel (Gráfico 4), esto podría ser ya que el orden de condición del problema es muy alto ($K(A) \gg 1$).

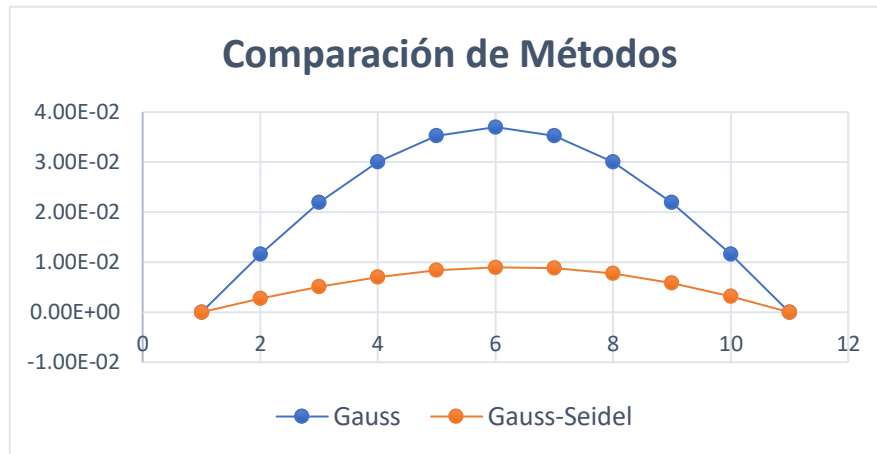


Gráfico 4: Gauss y Gauss-Seidel 10 nodos

No podemos comparar el método de Jacobi con ninguno de los otros dos ya que este no converge.

5. Anexo:

Códigos

Notas preliminares para lectura:

Para facilitar la resolución de los problemas, las siguientes variables fueron declaradas variables globales, para que en los distintos casos solo haya que reemplazar su valor:

```
L = 1; #Largo de la viga
nodos = 10; #Cantidad de nodos
E = 1; #Modulo de Young
I = 1; #Factor de Inercia
rtol=0.01; #Tolerancia
x0=zeros(nodos+1,1); #X0 = 0
```

Código 1: Sistema de Ecuaciones Lineales.

Recibe la cantidad de nodos y devuelve una matriz que cumpla las ecuaciones planteadas en Metodología.

```
function matriz = calcular_matriz(nodos)
matriz = [];
ceros = zeros(1, nodos);
agregar = [1 ceros]; #i == 0
matriz = [matriz agregar];

for i = 1 : nodos;
    if (i == 1);
        ceros = zeros(1, nodos - 3);
        agregar = [-4 5 -4 1 ceros];
        matriz = [matriz ; agregar];

    elseif (1 < i && i < nodos-1);
        cantidad_ceros_adelante = i - 2;
        ceros_adelante = zeros(1, cantidad_ceros_adelante);
        ceros_final = zeros (1, nodos - 4 -
        cantidad_ceros_adelante);

        agregar = [ceros_adelante 1 -4 6 -4 1 ceros_final];
        matriz = [matriz ; agregar];

    elseif (i == nodos-1);
        ceros = zeros(1, nodos - 3);
        agregar = [ceros 1 -4 5 -4];
        matriz = [matriz ; agregar];

    else
        ceros = zeros(1, nodos);
        agregar = [ceros 1];
        matriz = [matriz; agregar];
    endif
endfor
endfunction
```

Para mayor comprensión se decidió hacer en funciones aparte el cálculo de f y de la función de carga:

Función auxiliar 1:

Recibe el número de fila actual de f, el largo de la viga, el módulo de Young, el momento de inercia y la cantidad de nodos y calcula f. Corresponde a la **Ecuación 1**.

```
function f = calcular_f (i, L,E,I, nodos)
    f = calcular_funcion_carga(i, L, nodos) / (E * I)
    f = f * (L/nodos)^4
endfunction
```

Función auxiliar 2:

Recibe el número de fila actual, el largo de la viga y la cantidad de nodos y calcula la carga. Corresponde a la **Ecuación 2**.

```
function q = calcular_funcion_carga (i, L, nodos)
    x_i = i * L / nodos
    q = 2 + 4 * (x_i - x_i^2)
endfunction
```

Código 2: Sistema de Ecuaciones Lineales, resultado.

Recibe la cantidad de nodos y devuelve una matriz que cumpla las ecuaciones planteadas en Metodología.

```
function resultado = calcular_matriz_resultado (nodos, L,E,I)
    resultado = [0];
    for i = 1 : nodos - 1;
        resultado = [resultado; calcular_f(i, L, E, I, nodos)]
    endfor
    resultado = [resultado; 0]
endfunction
```

Código 3: Eliminación Gaussiana

Recibe una matriz que contenga las ecuaciones lineales, y otra que cumpla con estas ecuaciones, y devuelve la solución.

```
function R=Gauss(m,b)
    W=[m b];
    F=rows(W);
    C=columns(W);
    V=(1:F)';
    W=[W V];
    #Triangulamos la matriz
    for i=1:F
        wt=W(i:F,i:(C+1));
        mwt=abs(wt(:,1));
        [M,p]=max(mwt); #se realiza pivoteo si es necesario
        x=wt(1,:);
        wt(1,:)=wt(p,:);
        wt(p,:)=x;
        f=rows(wt);
        c=columns(wt);
        for j=2:f
            m=wt(j,1)/wt(1,1);
            wt(j,1:(c-1))=wt(j,1:(c-1))-m*(wt(1,1:(c-1)));
            #wt(j,1)=m;
        endfor
        W(i:F,i:(C+1))=wt;
    endfor
    W(:,C+1)=[]
    B=W(:,C)
    W(:,C)=[]
    R=W\B
endfunction
```

Código 4: Jacobi

Recibe la tolerancia, un x_0 , una matriz que contenga las ecuaciones lineales, y otra que cumpla con estas ecuaciones

```
function [x, iter, p] = jacobi (A, b, x0, tol)
    n = length(b);
    x = x0;
    iter = 0;
    err=tol+1;
    p=0;
    while err > tol
        x_old = x;
        for i = 1:n
            #Apliación Fórmula Jacobi
            x(i) = (b(i) - A(i,1:i-1)*x_old(1:i-1) -
A(i,i+1:n)*x_old(i+1:n)) / A(i,i);
        end
        iter = iter + 1;
        err = norm(x - x_old) / norm(x);
    end
    #Cálculo convergencia
    e = zeros(iter-1, 1);
    for k = 2:iter
        e(k-1) = norm(x - x_old) / norm(x_old - x0)^k;
    end
    p = log(e(2:end)./e(1:end-1)) / log(norm(x - x_old) /
norm(x_old - x0));
    p=p(3,1);
end
```

Función auxiliar 1: Cálculo de T y C

```
function [T, c] = jacobi_iteration_matrix(A, b)
    n = size(A, 1);
    T = zeros(n, n);
    c = zeros(n, 1);
    for i = 1:n
        d = A(i, i);
        T(i, i) = 0;
        for j = 1:n
            if i == j
                continue;
            endif
            T(i, j) = -A(i, j) / d;
        endfor
        c(i) = b(i) / d;
    endfor
endfunction
```

Función auxiliar 2: radio espectral de T

Recibe una matriz, y devuelve su radio espectral

```
function [r] = spectral_radius (A)

    e = eig(A);
    r = max(abs(e));
endfunction
```

Código 5: Gauss-Seidel

Recibe la tolerancia, un x_0 , una matriz que contenga las ecuaciones lineales, y otra que cumpla con estas ecuaciones, y devuelve la solución, el número de iteraciones y el orden de convergencia.

```
function [x, iter, p] = gauss_seidel(A, b, x0, tol)
    #Tomamos el largo de b para saber cantidad de ecuaciones
    n = length(b);
    x = x0;
    iter = 0;
    err = tol + 1;
    p=0;
    while err > tol
        x_old = x;
        for i = 1:n
            #Fórmula Gauss-Seidel
            x(i) = (b(i) - A(i,1:i-1)*x(1:i-1) -
A(i,i+1:n)*x_old(i+1:n)) / A(i,i);
        end
        iter = iter + 1;
        err = norm(x - x_old) / norm(x);
    end
    # Cálculo convergencia
    e = zeros(iter-1, 1);
    for k = 2:iter
        e(k-1) = norm(x - x_old) / norm(x_old - x0)^k;
    end
    p = log(e(2:end)./e(1:end-1)) / log(norm(x - x_old) /
norm(x_old - x0));
    p=p (3,1);
end
```

Función auxiliar 1: Cálculo de T y C

```
function [T, c] = gauss_seidel_iteration_matrix(A, b)
    n = size(A, 1);
    T = zeros(n, n);
    c = zeros(n, 1);
    for i = 1:n
        d = A(i, i);
        T(i, i) = 0;
        for j = 1:n
            if j <= i
                T(i, j) = -A(i, j) / d;
            else
                T(i, j) = -A(i, j);
            endif
        endfor
        c(i) = b(i) / d;
    endfor
endfunction
```

Resultados de Deformación de la Viga

Tabla n Resultados para 10 nodos, RTOL = 0,01, EI = 1

Jacobi	Gauss	Gauss - Seidel
0	-1.91E-17	0
-1.81E+307	0.011583	0.002694
2.66E+307	0.021936	0.005103
-Inf	0.030065	0.007038
Inf	0.03524	0.008365
-Inf	0.037015	0.008975
Inf	0.03524	0.008783
-Inf	0.030065	0.007731
2.66E+307	0.021936	0.005821
-1.81E+307	0.011583	0.003158
0	0	0

Tabla n Resultados para 50 nodos, RTOL = 0,01, EI = 1

Gauss	Gauss - Seidel	Jacobi
-2.30E-17	0	0
0.002333	3.34E-06	3.86E+306
0.004655	6.32E-06	-5.95E+306
0.006956	8.76E-06	8.31E+306
0.009227	1.06E-05	-1.06E+307
0.011458	1.20E-05	1.28E+307
0.01364	1.29E-05	-1.50E+307
0.015764	1.35E-05	1.72E+307
0.017822	1.39E-05	-1.93E+307
0.019806	1.41E-05	2.13E+307
0.021708	1.43E-05	-2.32E+307
0.023522	1.44E-05	2.50E+307
0.02524	1.46E-05	-2.68E+307
0.026856	1.47E-05	2.84E+307
0.028365	1.49E-05	-3.00E+307
0.029761	1.50E-05	Inf
0.031038	1.52E-05	-inf
0.032194	1.53E-05	Inf
0.033223	1.55E-05	-inf
0.034122	1.56E-05	Inf
0.034888	1.57E-05	-inf
0.035519	1.58E-05	Inf
0.036011	1.59E-05	-inf
0.036364	1.59E-05	Inf
0.036576	1.60E-05	-inf
0.036647	1.60E-05	Inf
0.036576	1.60E-05	-inf

Gauss	Gauss - Seidel	Jacobi
0.036364	1.60E-05	Inf
0.036011	1.60E-05	-inf
0.035519	1.59E-05	Inf
0.034888	1.59E-05	-inf
0.034122	1.58E-05	Inf
0.033223	1.57E-05	-inf
0.032194	1.55E-05	Inf
0.031038	1.54E-05	-inf
0.029761	1.52E-05	Inf
0.028365	1.50E-05	-3.00E+307
0.026856	1.48E-05	2.84E+307
0.02524	1.46E-05	-2.68E+307
0.023522	1.45E-05	2.50E+307
0.021708	1.45E-05	-2.32E+307
0.019806	1.45E-05	2.13E+307
0.017822	1.46E-05	-1.93E+307
0.015764	1.46E-05	1.72E+307
0.01364	1.43E-05	-1.50E+307
0.011458	1.38E-05	1.28E+307
0.009227	1.26E-05	-1.06E+307
0.006956	1.06E-05	8.31E+306
0.004655	7.85E-06	-5.95E+306
0.002333	4.22E-06	3.86E+306
0	0	0

Tabla n Resultados para 100 nodos, RTOL = 0,01, EI = 1

Se eligió mostrar de forma gráfica debido a la gran cantidad de datos.

