ED con CDF de orden superior

Curso de Física Computacional

M. en C. Gustavo Contreras Mayén

Facultad de Ciencias - UNAM

25 de abril de 2018

1. ED con CDF de orden superior

2. Funciones para matrices en banda

ED de cuarto orden

Hemos estado revisando la conexión que hay entre el tema anterior (ED con CDF) con el tema de álgebra matricial.

Dos temas unidos

Contamos ya con las herramientas conceptuales así como de solución numérica, para abordar un problema que plantea un sistema algebraico donde la expresión matrical, será de gran ayuda.

Dos temas unidos

Pondremos en marcha la solución de ese sistema, con las herramientas que nos brinda `python`.

ED de orden superior

En aras de la brevedad, limitamos nuestra discusión al caso especial en el que las derivadas y', así como y''' no aparecen explícitamente en la ecuación diferencial; es decir, consideramos

$$y^{(4)} = f(x, y, y'')$$

Características de la ED

Suponemos que las dos CDF se presentan en cada extremo del dominio de solución (a, b).

Los problemas de esta tipo se encuentran comúnmente en la teoría de vigas.

Solución de la ED

Nuevamente, dividimos el dominio de solución en m intervalos de longitud h cada uno.

Sustituyendo las derivadas de y por diferencias finitas en los puntos de malla, obtenemos las ecuaciones de diferencias:

Ecs. en diferencias finitas

$$\begin{aligned}\frac{y_{i-2} - 4\,y_{i-1} + 6\,y_i - 4\,y_{i+1} + y_{i+2}}{h^4} = \\ = f\left(x_i, y_i, \frac{y_{i-1} - 2\,y_i + y_{i+1}}{h^2}\right)\end{aligned} \tag{1}$$

donde $i = 0, 1, \ldots, m$

Re-escribiendo las ecuaciones

$$\begin{multline} y_{-2} - 4\,y_{-1} + 6\,y_0 - 4\,y_1 + y_2 + \\ - h^4\, f\left(x_0, y_0, \frac{y_{-1} - 2\,y_0 + y_1}{h^2}\right) = 0 \tag{2} \end{multline}$$

$$\begin{multline} y_{-1} - 4\,y_0 + 6\,y_1 - 4\,y_2 + y_3 + \\ - h^4\, f\left(x_1, y_1, \frac{y_0 - 2\,y_1 + y_2}{h^2}\right) = 0 \tag{3} \end{multline}$$

$$\begin{multline} y_0 - 4\,y_1 + 6\,y_2 - 4\,y_3 + y_4 + \\ - h^4\, f\left(x_2, y_2, \frac{y_1 - 2\,y_2 + y_3}{h^2}\right) = 0 \tag{4} \end{multline}$$

$$\vdots$$

Re-escribiendo las ecuaciones

$$\vdots$$

$$\begin{aligned} y_{m-3} - 4\,y_{m-2} + 6\,y_{m-1} - 4\,y_m + y_{m+1} + \\ - h^4\,f\left(x_{m-1}, y_{m-1}, \frac{y_{m-2} - 2\,y_{m-1} + y_m}{h^2}\right) = 0 \end{aligned} \tag{5}$$

$$\begin{aligned} y_{m-2} - 4\,y_{m-1} + 6\,y_m - 4\,y_{m+1} + y_{m+2} + \\ - h^4\,f\left(x_m, y_m, \frac{y_{m-1} - 2\,y_m + y_{m+1}}{h^2}\right) = 0 \end{aligned} \tag{6}$$

Uso de las condiciones de frontera

Vemos que hay cuatro incógnitas

$$y_2, y_1, y_{m+1}, y_{m+2}$$

que se encuentran fuera del dominio de la solución y deben eliminarse aplicando las condiciones de frontera.

Para ello, nos apoyamos con la siguiente tabla:

Condiciones de frontera para α

CDF	Expresión equivalente de Dif. Fin.
$y(a) = \alpha$	$y_0 = \alpha$
$y'(a) = \alpha$	$y_{-1} = y_1 - 2\,h\,\alpha$
$y''(a) = \alpha$	$y_{-1} = 2\,y_0 - y_1 + h^2\,\alpha$
$y'''(a) = \alpha$	$y_{-2} = 2\,y_1 - 2\,y_1 + y_2 - 2\,h^3\,\alpha$

Condiciones de frontera para β

CDF	Expresión equivalente de Dif. Fin.
$y(b) = \beta$	$y_m = \beta$
$y'(b) = \beta$	$y_{m+1} = y_{m-1} - 2\,h\,\beta$
$y''(b) = \beta$	$y_{m+1} = 2\,y_m - y_{m-1} + h^2\,\beta$
$y'''(b) = \beta$	$y_{m+2} = 2\,y_{m+1} - 2\,y_{m-1} + y_{m-2} - 2\,h^3\,\beta$

Sobre las CDF

Mirando detenidamente, se puede notar que algunas combinaciones de las CDF no funcionarán para eliminar el “exceso”.

Sobre las CDF

Mirando detenidamente, se puede notar que algunas combinaciones de las CDF no funcionarán para eliminar el “exceso”.

Una de estas combinaciones es claramente $y(a) = \alpha_1$ y $y'''(a) = \alpha_2$.

Sobre las CDF

Mirando detenidamente, se puede notar que algunas combinaciones de las CDF no funcionarán para eliminar el “exceso”.

Una de estas combinaciones es claramente $y(a) = \alpha_1$ y $y'''(a) = \alpha_2$.

La otra condición es $y'(a) = \alpha_1$ y $y''(a) = \alpha_2$.

Sobre las CDF

En el contexto de la teoría de vigas, esto tiene sentido: podemos imponer un desplazamiento y o una fuerza de corte $E\,I\,y'''$ en un punto, *pero es imposible aplicar ambos simultáneamente*.

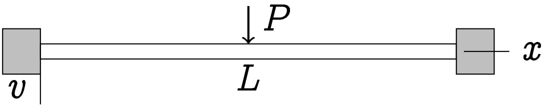
Sobre las CDF

De manera similar, no tiene sentido físico indicar tanto la pendiente y' como el momento de flexión $E\ I\ y''$ en el mismo punto.

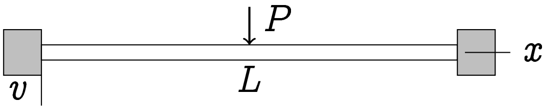
Ejercicio

Una viga uniforme de longitud L y con rigidez a la flexión $E\,I$ está unida en los extremos mediante soportes rígidos.

La viga lleva una carga concentrada P en su punto medio.



Ejemplo



Si utilizamos la simetría y modelamos sólo la mitad izquierda de la viga, resolvemos la siguiente ED con CDF y obtenemos el desplazamiento v:

ED del problema de la viga

$$E\,I\,\frac{d^4 v}{dx^4} = 0$$

Con las CDF

$$v\big|_{x=0} = 0, \qquad \left.\frac{dv}{dx}\right|_{x=0} = 0$$

$$\left.\frac{dv}{dx}\right|_{x=L/2} = 0, \qquad E\,I\,\left.\frac{d^3 v}{dx^3}\right|_{x=L/2} = -\frac{P}{2}$$

Solución

Usaremos el método de diferencias finitas para estimar el desplazamiento y el momento de flexión en el punto medio:

$$M = -E\, I \frac{d^2 v}{dx^2}$$

Los valores exactos de la solución son:

1. $v = \dfrac{P\, L^3}{192\, E\, I}$

Solución

Usaremos el método de diferencias finitas para estimar el desplazamiento y el momento de flexión en el punto medio:

$$M = -E\,I\,\frac{d^2 v}{dx^2}$$

Los valores exactos de la solución son:

1. $v = \dfrac{P\,L^3}{192\,E\,I}$
2. $M = \dfrac{P\,L}{8}$

Solución

Hacemos el siguiente cambio de variable

$$\xi = \frac{x}{L} \qquad y = \frac{E\,I}{P\,L^3}\,v$$

Solución

Hacemos el siguiente cambio de variable

$$\xi = \frac{x}{L} \qquad y = \frac{E\,I}{P\,L^3}\,v$$

entonces, tendremos el siguiente problema:

$$\frac{d^4 y}{d\xi^4} = 0$$

Con las CDF

$$y\big|_{\xi=0} = 0, \qquad \left.\frac{dy}{d\xi}\right|_{\xi=0} = 0$$

$$\left.\frac{dy}{d\xi}\right|_{\xi=1/2} = 0, \qquad \left.\frac{d^3 y}{d\xi^3}\right|_{\xi=1/2} = -\frac{1}{2}$$

Ecs. en diferencias finitas

Escribimos las ecuaciones (2) - (6), considerando las CDF, de acuerdo a la tabla anterior, las condiciones en el extremo izquierdo son: $y_0 = 0$ y $y_{-1} = y_1$.

Ecs. en diferencias finitas

De aquí que las ecs. (2) y (3) son:

$$y = 0 \tag{7}$$

$$-4\,y_0 + 7\,y_1 - 4\,y_2 + y_3 = 0 \tag{8}$$

Ecs. en diferencias finitas

De aquí que las ecs. (2) y (3) son:

$$y = 0 \tag{7}$$

$$-4\,y_0 + 7\,y_1 - 4\,y_2 + y_3 = 0 \tag{8}$$

La ec. (4) es

$$y_0 - 4\,y_1 + 6\,y_2 - 4\,y_3 + y_4 = 0 \tag{9}$$

Ecs. en diferencias finitas

En el extremo derecho, las CDF son equivalentes a $y_{m+1} = y_{m-1}$, junto con

$$y_{m+2} = 2y_{m+1} + y_{m-2} - 2y_{m-1} + 2h^3(-1/2) = y_{m-2} - h^3$$

Ecs. en diferencias finitas

En el extremo derecho, las CDF son equivalentes a $y_{m+1} = y_{m-1}$, junto con

$$y_{m+2} = 2y_{m+1} + y_{m-2} - 2y_{m-1} + 2h^3(-1/2) = y_{m-2} - h^3$$

Al sustituir en las ecuaciones (5) y (6), llegamos a

$$y_{m-3} - 4\, y_{m-2} + 7\, y_{m-1} - 4\, y_m = 0 \qquad (10)$$

$$2\, y_{m-2} - 8\, y_{m-1} + 6\, y_m = h^3 \qquad (11)$$

Configuración de la matriz

La matriz de coeficientes de las ecs. (7) - (11) se puede hacer simétrica, al dividir la ec. (11) entre 2.

Así obtenemos el siguente sistema algebraico:

Sistema matricial del problema

$$
\begin{bmatrix}
1 & 0 & 0 & & & & \\
0 & 7 & -4 & 1 & & & \\
0 & -4 & 6 & -4 & 1 & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & 1 & -4 & 6 & -4 & 1 \\
 & & & 1 & -4 & 7 & -4 \\
 & & & & 1 & -4 & 3
\end{bmatrix}
\begin{bmatrix}
y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{m-2} \\ y_{m-1} \\ y_m
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0.5\, h^3
\end{bmatrix}
$$

Usando `scipy.solve_banded`

Ahora nos podemos apoyar con la función **`solve_banded`**, que está incluida en **`scipy`**.

Para ello, debemos de construir los vectores columna necesarios, a partir de las codiagonales y de la diagonal principal.

Construcción de los vectores

Con la función **`diagonales`** se construyen los vectores para el arreglo **`cm`** -matriz de coeficientes- que requiere la función **`solve_banded`**:

Función `diagonales` I

Código 1: Función que construye las codiagonales

```
def diagonales(x, h, m):
    d = np.ones(m) * 6.0
    e = np.ones(m) * (-4.0)
    f = np.ones(m)
    c = e.copy()
    g = f.copy()
    b = np.zeros(m)

    d[0] = 1.0
    d[1] = 7.0
    d[-1] = 3.0
    d[-2] = 7.0
```

Función diagonales II

```

    e[0] = 0.0
    e[1] = 0.0

    f[0] = 0.0
    f[1] = 0.0
    f[2] = 0.0

    c[0] = 0.0
    c[-1] = 0.0

    g[0] = 0.0
    g[-1] = 0.0
```

Función diagonales III

```
    g[-2] = 0.0

    b[-1] = 0.5 * h**3
    return f, e, d, c, g, b
```

Geometría de la viga

Código 2: Geometría de la viga

```
 # x en el extremo izquierdo
xInicio = 0.0

# x en el extremo derecho
xFinal = 0.5

# numero de puntos en la malla
m = 20

h = (xFinal - xInicio)/m

x = np.arange(xInicio, xFinal+h, h)
```

Uso de las codiagonales

Código 3: Uso de las codiagonales

```
f, e, d, c, g, b = diagonales(x, h,
   m+1)

cm = np.array([f, e, d, c, g])

sol = scipy.linalg.solve_banded((2,
   2), cm, b)
```

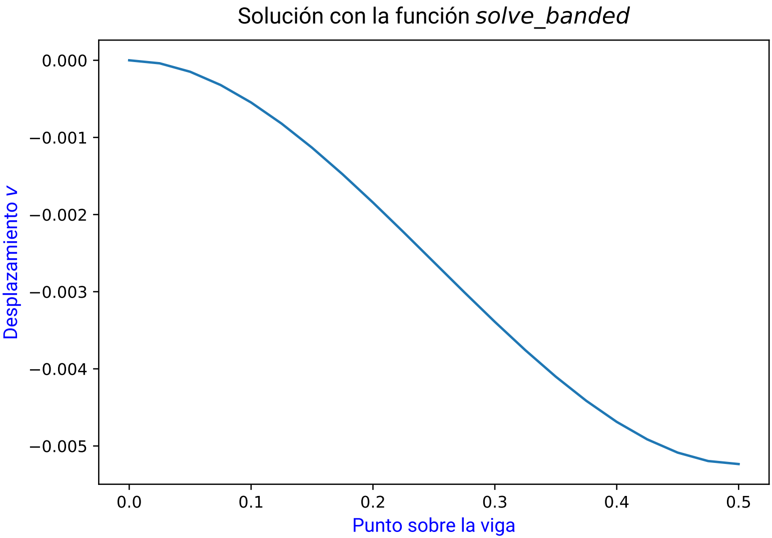
Rutina de impresión

Código 4: Impresión de los resultados

```python
print('punto \t \t desplazamiento')
print('-'*35)

for i in range(len(sol)):
    print('{:1.3e} \t {:1.5e}'.format(x[i], sol[i]))
```

Gráfica de la solución



Análisis de la solución

Habiendo ejecutado el programa con $m = 20$, tomamos las dos últimas líneas de la solución:

punto	x	y
$m-1$	$4.75000e-01$	$5.19531e-03$
m	$5.00000e-01$	$5.23438e-03$

Análisis de la solución

Por lo tanto, a la mitad de la viga tenemos que:

$$v\big|_{x=0.5\,L} = \frac{P\,L^3}{E\,I}\,y\Big|_{\xi=0.5} = 5.23438 \times 10^{-3}\,\frac{P\,L^3}{E\,I}$$

Análisis de la solución

Por lo tanto, a la mitad de la viga tenemos que:

$$v\Big|_{x=0.5\,L} = \frac{P\,L^3}{E\,I}\,y\Big|_{\xi=0.5} = 5.23438 \times 10^{-3}\,\frac{P\,L^3}{E\,I}$$

La solución exacta es:

$$v\Big|_{x=0.5\,L} = 5.20833 \times 10^{-3}\,\frac{P\,L^3}{E\,I}$$

Análisis de la solución

Usamos los valores $m - 1$ y m de la solución que recuperamos con el algoritmo:

$$\begin{aligned}
\left.\frac{d^2 v}{dx^2}\right|_{x=0.5\,L} &= \frac{P\,L^3}{E\,I}\left(\frac{1}{L^2}\left.\frac{d^2 y}{d\xi^2}\right|_{\xi=0.5}\right) \\
&\simeq \frac{P\,L}{E\,I}\,\frac{y_{m-1} - 2y_m + y_{m+1}}{h^2} \\
&= \frac{P\,L}{E\,I}\,\frac{(5.19531 - 2(5.23438) + 5.19531) \times 10^{-3}}{h^2} \\
&= -0.125024\,\frac{P\,L}{E\,I}
\end{aligned}$$

Análisis de la solución

Ya podemos calcular el momento de flexión de la viga:

$$M\Big|_{x=0.5\,L} = -E\,I\,\frac{d^2v}{dx^2}\Bigg|_{\xi=0.5} = 0.125024\,P\,L$$

Análisis de la solución

Ya podemos calcular el momento de flexión de la viga:

$$M\Big|_{x=0.5\ L} = -E\ I\ \frac{d^2 v}{dx^2}\Bigg|_{\xi=0.5} = 0.125024\ P\ L$$

La solución exacta es:

$$M\Big|_{x=0.5\ L} = 0.125000\ P\ L$$

Gráfica completa de la solución

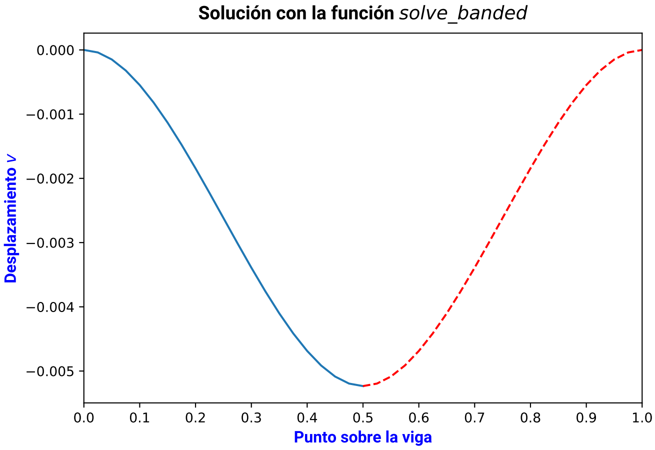


Figura 1: *El otro extremo de la viga tiene por simetría, los mismos valores en la solución, sólo invierte el orden de los datos de* `sol`.

Uso de funciones para matrices en banda

Desarrollamos dos problemas que nos condujeron a un sistema tridiagonal y a otro pentadiagonal.

La elaboración de funciones para factorizar mediante LU y luego resolver la factorización, para cada caso, como vimos es un proceso que sólo funciona en cada problema.

Ventajas con `scipy`

Usar la función **`solve_banded`**, no implica complicación mayor (uso de los argumentos), el punto crítico es construir la matriz en banda, mediante el análisis y trabajo propiamente en el cuaderno.

¿Hay más del tema?

De acuerdo a la orientación que decidan llevar como físicos, encontrarán problemas que les demanden el manejo de soluciones con matrices, lo que se ha revisado, es lo general pero no lo único.

El tema se puede extender más, pero ya enfocado en particular a lo que deban de resolver. Cuentan ya con los elementos básicos para aplicarlo en su formación.

Considera la siguiente matriz en banda

$$\begin{bmatrix} 1 & -5 & 3 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & -3 & 5 & 0 & 0 & 0 & 0 \\ 0 & -2 & 1 & 5 & 9 & 0 & 0 & 0 \\ 0 & 0 & 9 & -1 & 5 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -3 & -2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 & -6 \\ 0 & 0 & 0 & 0 & 0 & -3 & 2 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9 & 1 \end{bmatrix}$$

Marcando la diagonal y las codiagonales

$$\begin{pmatrix} 1 & -5 & 3 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & -3 & 5 & 0 & 0 & 0 & 0 \\ 0 & -2 & 1 & 5 & 9 & 0 & 0 & 0 \\ 0 & 0 & 9 & -1 & 5 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -3 & -2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 & -6 \\ 0 & 0 & 0 & 0 & 0 & -3 & 2 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9 & 1 \end{pmatrix}$$

Arreglo de las codiagonales

$$\begin{pmatrix} 0 & 0 & \boxed{3 & 5 & 9 & 4 & -2 & -6} \\ 0 & \boxed{-5 & -3 & 5 & 5 & -3 & 1 & 7} \\ \boxed{1 & 2 & 1 & -1 & 2 & 0 & 2 & 1} \\ \boxed{3 & -2 & 9 & 0 & 2 & -3 & 9} & 0 \end{pmatrix}$$

- Las codiagonales superiores tienen ceros a la izquierda.
- Las codiagonales inferiores tienen ceros al final.

Nueva representación de la matriz

La matriz en banda se puede representar por una matriz no cuadrada del tipo

Nueva representación de la matriz

La matriz en banda se puede representar por una matriz no cuadrada del tipo

$$
\begin{array}{c}
g \\ c \\ d \\ f
\end{array}
\begin{pmatrix}
0 & 0 & 3 & 5 & 9 & 4 & -2 & -6 \\
0 & -5 & -3 & 5 & 5 & -3 & 1 & 7 \\
1 & 2 & 1 & -1 & 2 & 0 & 2 & 1 \\
3 & -2 & 9 & 0 & 2 & -3 & 9 & 0
\end{pmatrix}
$$

Resolviendo matrices en banda

Para resolver un sistema de ecuaciones con una matriz en banda, usaremos la función **`scipy.linalg.solve_banded()`**.

Resolviendo matrices en banda

Para resolver un sistema de ecuaciones con una matriz en banda, usaremos la función **`scipy.linalg.solve_banded()`**.

La sintaxis para esta función es la siguiente

```
solve_banded((l,u), cm, rhs)
```

Resolviendo matrices en banda

`solve_banded((l,u), cm, rhs)`

Donde:

- **`(l, u)`** es una tupla, donde l es el número de codiagonales inferiores no nulas, y u es el número de codiagonales superiores no nulas.

Resolviendo matrices en banda

`solve_banded((l,u), cm, rhs)`

Donde:

- `(l, u)` es una tupla, donde l es el número de codiagonales inferiores no nulas, y u es el número de codiagonales superiores no nulas.
- `cm` es la matriz en banda de coeficientes.

Resolviendo matrices en banda

`solve_banded((l,u), cm, rhs)`

Donde:

- **`(l, u)`** es una tupla, donde l es el número de codiagonales inferiores no nulas, y u es el número de codiagonales superiores no nulas.
- **`cm`** es la matriz en banda de coeficientes.
- **`rhs`** es el vector de constantes del lado derecho.

Ejercicio 1 - Matriz en banda

Resuelve el siguiente sistema algebraico

$$\begin{bmatrix} 1 & -5 & 3 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & -3 & 5 & 0 & 0 & 0 & 0 \\ 0 & -2 & 1 & 5 & 9 & 0 & 0 & 0 \\ 0 & 0 & 9 & -1 & 5 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -3 & -2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 & -6 \\ 0 & 0 & 0 & 0 & 0 & -3 & 2 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ -3 \\ 2 \\ 0 \\ 7 \\ 8 \\ 1 \end{bmatrix}$$

Solución al sistema

$$
\begin{aligned}
x_1 &= 200.639937 \\
x_2 &= -25.491352 \\
x_3 &= -107.698899 \\
x_4 &= -174.206761 \\
x_5 &= 102.750000 \\
x_6 &= 70.833333 \\
x_7 &= -3.500000 \\
x_8 &= 32.500000
\end{aligned}
$$

Orden de los vectores renglón

Hay que tener cuidado al momento de organizar los vectores renglón que conformarán el arreglo `cm`:

Código 5: Definición de los vectores renglón

```
g = [0. ,0, 3, 5, 9, 4, -2, -6]
c = [0., -5, -3, 5, 5, -3, 1, 7]
d = [1., 2, 1, -1, 2, 0, 2, 1]
e = [3., -2, 9, 0, 2, -3, 9, 0]

b = np.array([5., 3, -3, 2, 0, 7, 8, 1])

cm = np.array([g, c, d, e])
```

Solución al problema

Una vez que definimos el orden de los vectores renglón, ya podemos usar la función `solve_banded` para encontrar la solución.

Uso de una librería de formato

Cuando tenemos que presentar en la terminal un conjunto de datos, en ocasiones invertimos más tiempo en ajustar el formato de salida: dejando espacios, tabuladores, etc.

La librería Texttable

Usaremos la librería `Texttable` para ahorrar tiempo y dejar una salida en la terminal mucho más amigable.

La librería Texttable

Primer paso: Abre una terminal de comandos, en Linux con la combinación Control + Mayus + T

Escribe la instrucción:

```
pip install texttable
```

La librería Texttable

Primer paso: Abre una terminal de comandos, en Linux con la combinación Control + Mayus + T

Escribe la instrucción:

```
pip install texttable
```

Esto nos dejará la librería para su uso global.

En tu código

Segundo paso: Haremos una modificación en el código para mostrar los resultados en la terminal con la nueva librería `Texttable`.

Código para elaborar la tabla

En el apartado de encabezados, donde se importan las librerías:

Código 6: Importando la librería Texttable

```
import texttable as tt
```

Código para elaborar la tabla

Código 7: Creando el objeto y definiendo propiedades

```
tabla = tt.Texttable()

x = [[]]

for i in range(8):
    x.append(['x' + str(i+1), sol[i]])

tabla.add_rows(x)
tabla.header(['x', 'Solucion'])
tabla.set_cols_align(["c", "l"])
print(tabla.draw())
```

Salida en la terminal

```
+----+-----------+
| x  | Solución  |
+====+===========+
| x1 | 200.640   |
+----+-----------+
| x2 | -25.491   |
+----+-----------+
| x3 | -107.699  |
+----+-----------+
| .. |    ...    |
+----+-----------+
| x7 | -3.500    |
+----+-----------+
| x8 | 32.500    |
+----+-----------+
```

Tabla con formato

Como podrás ver, la tabla que se muesta en la terminal ajusta los contenidos a partir de las propiedades de alineación (`['c', 'l']`), ajusta hacia el centro y a la izquierda.

En la documentación de la librería podrás encontrar más referencias para ajustar tu salida en la terminal con una presentación más profesional.