

Implementar el Método de Elementos Finitos (MEF) en Python

Implementar el Método de Elementos Finitos (MEF) en Python puede ser un proceso detallado, pero aquí te dejo una guía básica para que puedas empezar:

Pasos para Implementar el MEF en Python

1. Instalación de Bibliotecas Necesarias

- Asegúrate de tener instaladas las bibliotecas necesarias como NumPy y SciPy. Puedes instalarlas usando pip:

```
>> pip install numpy scipy
```

2. Definición del Problema

- Define el problema que deseas resolver. Por ejemplo, resolver la ecuación de Schrödinger para una partícula en una caja.

3. Discretización del Dominio

- Divide el dominio del problema en elementos finitos. Esto implica crear una malla que divida el espacio en pequeños elementos.

```
import numpy as np
```

```
# Definir el dominio
```

```
L = 1.0 # Longitud de la caja
```

```
n_elements = 10 # Número de elementos
```

```
x = np.linspace(0, L, n_elements + 1) # Puntos nodales
```

4.

5. Formulación de la Matriz de Rigidez y el Vector de Fuerzas

- Calcula la matriz de rigidez y el vector de fuerzas para cada elemento y ensámblalos en matrices globales.

```
K = np.zeros((n_elements + 1, n_elements + 1)) # Matriz de rigidez global
```

```
F = np.zeros(n_elements + 1) # Vector de fuerzas global
```

```
# Ensamblar la matriz de rigidez y el vector de fuerzas
```

```
for i in range(n_elements):
```

```
    k_local = np.array([[1, -1], [-1, 1]]) # Matriz de rigidez local
```

```
    f_local = np.array([0, 0]) # Vector de fuerzas local
```

```
# Ensamblar en la matriz global
```

```
K[i:i+2, i:i+2] += k_local
```

```
F[i:i+2] += f_local
```

6.

7. Aplicación de Condiciones de Frontera

- Aplica las condiciones de frontera necesarias para tu problema.

```
# Condiciones de frontera (ejemplo:  $u(0) = 0$  y  $u(L) = 0$ )
```

```
K[0, :] = 0
K[:, 0] = 0
K[0, 0] = 1
F[0] = 0
```

```
K[-1, :] = 0
K[:, -1] = 0
K[-1, -1] = 1
F[-1] = 0
```

8.

9. Resolución del Sistema de Ecuaciones

- Resuelve el sistema de ecuaciones resultante para obtener la solución.

```
u = np.linalg.solve(K, F)
```

10.

11. Visualización de Resultados

- Utiliza bibliotecas como Matplotlib para visualizar los resultados.

```
import matplotlib.pyplot as plt
```

```
plt.plot(x, u, marker='o')
plt.xlabel('x')
plt.ylabel('u(x)')
plt.title('Solución de la Ecuación de Schrödinger')
plt.show()
```

12.

Recursos Adicionales

- Puedes encontrar ejemplos y más detalles en repositorios como [PMEF en GitHub\[1\]](#).
- Cursos y tutoriales en línea también pueden ser de gran ayuda para profundizar en la implementación del MEF en Python[2].

References

[1] [Programando el Método de Elementos Finitos en Python](#)

[2] [Método dos Elementos Finitos Em Python: Aprenda a Simular ... - Awari](#)