Análisis numérico 2025-2S
Tarea 1 (Grupo 1)

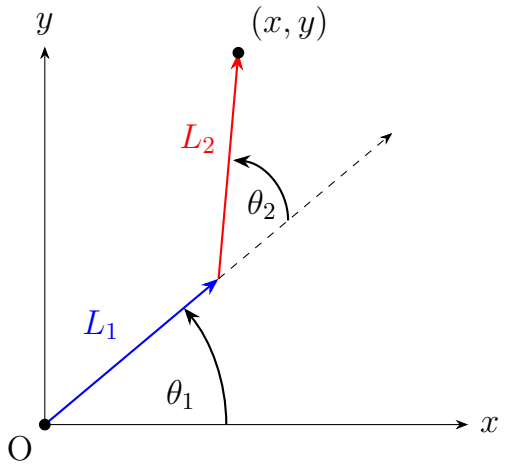
Parte 1: Solución de sistemas no lineales

La cinemática inversa de un brazo robótico

La robótica moderna requiere que los sistemas autónomos realicen tareas con precisión. Uno de los problemas fundamentales es la cinemática inversa, es decir, determinar los ángulos que deben adoptar las articulaciones de un brazo robótico para alcanzar una posición deseada en el plano (o el espacio). Este problema da origen a sistemas de ecuaciones no lineales, cuya solución requiere métodos numéricos adecuados.

Descripción del problema

Considere un brazo robótico planar compuesto por dos eslabones de longitudes L_1 y L_2, articulados mediante ángulos θ_1 (el que forma el primer eslabón con la base de soporte) y θ_2 (el que forma el segundo eslabón respecto al primero.)



La posición del extremo del brazo en el plano (en coordenadas cartesianas) está dada por:

$$x(\theta_1, \theta_2) = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)$$

$$y(\theta_1, \theta_2) = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2)$$

Para alcanzar un punto deseado (x_d, y_d), se plantea el sistema no lineal:

$$F_1(\theta_1, \theta_2) = x(\theta_1, \theta_2) - x_d = 0$$
$$F_2(\theta_1, \theta_2) = y(\theta_1, \theta_2) - y_d = 0$$

Actividades

1. Implementar el método de Newton para resolver el sistema no lineal resultante.
2. Elegir valores realistas para L_1, L_2 y al menos tres posiciones destino (x_d, y_d) y evaluar los resultados del método.
3. Analizar la sensibilidad a las condiciones iniciales y discutir los casos en que el método converge y las dificultades en la implementación.
4. Visualizar las posiciones finales alcanzadas y las trayectorias intermedias del brazo.
5. Generalizar el modelo a un brazo con $n \geq 3$ eslabones. En este caso, el sistema se compone de n ecuaciones no lineales y el número de soluciones puede aumentar significativamente.

Entregable

El trabajo se debe desarrollar en grupos asignados previamente y consiste en la implementación de un método numérico en un entorno computacional. A continuación, se describen las condiciones y lineamientos para su correcta entrega:

- **Formato de entrega:** Cada grupo deberá entregar un único archivo `.ipynb` en formato `Jupyter Notebook`, que contenga todo el desarrollo del trabajo: formulación teórica, implementación, pruebas y análisis de resultados. Este debe estar bien estructurado, con celdas tipo Markdown explicando cada paso, y código comentado adecuadamente.
- **Lenguaje y estilo:** El notebook puede estar redactado en español o inglés, siempre que mantenga un tono claro y riguroso. Se valorará especialmente el uso adecuado del lenguaje matemático, figuras, y la claridad en la presentación de resultados.
- **Repositorio en GitHub:** La entrega debe realizarse mediante el repositorio del curso en Github

  `https://github.com/unalmedmathnum/numericalAnalysis_2025`
- **Participación individual:** Es obligatorio que todos los integrantes del grupo realicen al menos un *commit* individual al repositorio, donde se evidencie su participación activa en el desarrollo del código o del análisis.

- **Fecha de entrega:** La entrega se realizará de forma presencial durante el horario habitual de clase. Se deben presentar los resultados obtenidos ante sus compañeros, explicando brevemente el problema abordado, la metodología de solución, el análisis realizado y las conclusiones obtenidas. La presentación debe ser concisa y comprensible para todo el grupo, con el apoyo de visualizaciones o ejemplos.

Parte 2: Códigos de clase

Como parte de la tarea se deben implementar los siguientes métodos en archivos independientes, organizados y documentados.

- `propagacion_errores.py`: Cálculo de propagación del error en todas las operaciones básicas.
- `biseccion.py`: Método de bisección con criterios de tolerancia y máximo número de iteraciones.
- `punto_fijo.py`: Método del punto fijo con verificación de condiciones de convergencia.
- `newton.py`: Método de Newton con visualización de iteraciones y control de errores numéricos.
- `newton_modificado.py`: Método de Newton modificado con comparación frente a Newton.
- `secante.py`: Método de la secante con comparación frente a Newton.

Instrucciones de implementación

- Cada archivo `.py` debe contener al menos una función principal que reciba los parámetros clave del método y devuelva el resultado.
- El archivo debe finalizar con un bloque de ejemplo utilizando `if __name__ == "__main__"` para mostrar cómo se utiliza el método.
- El código debe estar debidamente comentado y documentado.
- La presentación de resultados se realizará de forma presencial en clase, donde el grupo explicará brevemente los métodos y mostrará sus resultados.

Sugerencia: Pueden guiarse por los códigos desarrollados el semestre pasado, disponibles públicamente en el siguiente repositorio de GitHub:

`https://github.com/unalmedmathnum/unalmedmathnum-analysis2024`