

TRABAJO PRÁCTICO N°1

Errores y Representación Numérica

Introducción

En los recipientes de presión de los reactores nucleares de agua pesada a presión (PHWR) como los que operan en Argentina, se utiliza la fisión (ruptura) de átomos de U238 por captura de neutrones térmicos (lentos). Cuando el U238 captura un neutrón, se rompe liberando más neutrones cuya velocidad disminuye al chocar con el agua del recipiente de presión. Después de sucesivos choques, los neutrones entregan energía al agua (que se calienta) y pierden velocidad, hasta que su velocidad es adecuada para que sean capturados por otros núcleos de U238.

Para ajustar la potencia a la que funciona el reactor, se utilizan barras de un material que absorbe los neutrones que se producen en la reacción nuclear. Si se absorben más neutrones de los que se producen, la reacción se frena (reacción sub-crítica). Si, por el contrario, se absorben menos neutrones, la reacción aumenta (reacción super-crítica). La cantidad de neutrones que las barras absorben se regula con la longitud de barra dentro del reactor, por lo que resulta de suma importancia conocer este dato en todo momento, así como contar con un mecanismo preciso de movimiento de las barras de control en el núcleo del reactor.

A los fines de esta práctica, vamos a suponer que el movimiento de las barras de control en el núcleo de un reactor nuclear está regulado por la siguiente pieza de código:

```
1 pos_inicial = 100;
2 pos_final = 5000;
3 delta = 0.01;
4 condicion = true;
5 i = 1;
6 pos_barra = pos_inicial;
7 while condicion
8     pos_barra = pos_inicial + i*delta;
9     i = i + 1;
10    if pos_barra >= pos_final
11        condicion = false;
12    endif
13 endwhile
14 disp(i);
15 disp(pos_barra);
```

1. Comprensión de código

- Lea el código anterior. Coméntelo. ¿Qué hace?
- Realice un diagrama de flujo representativo del código anterior.
- ¿Cumple el programa con su cometido? De no ser así, realice las modificaciones pertinentes. Realice primero el diagrama de flujo, y luego escriba el código.

2. Adaptación del código

- Modifique el programa para que pida, en forma interactiva, la posición inicial, la posición final y el paso de avance.
- Modifique el programa para que funcione con números de precisión simple.
- Analice el error que el programa comete al usar precisión simple (error absoluto, error relativo, cota del error). Grafique el error en función de la posición.

3. Problema inverso

Supongamos ahora que el error es de naturaleza física y no matemática. Es decir, que existe un error inherente al desplazamiento de las barras de control.

- ¿Cuáles son los errores que afectan la posición final de la barra?
- ¿Cuál es la cota del error de las variables involucradas para que la posición final quede determinada con un error máximo de 1mm?

4. Errores

- Se supondrá ahora que el motor tarda en alcanzar su velocidad máxima de desplazamiento, por lo que al arrancar (al recibir los primeros pulsos dentro de un ciclo de movimiento) acelera de acuerdo a su capacidad máxima. Determine el error en la posición, velocidad y la aceleración de la barra, si al motor de control se le pueden enviar hasta 500 pulsos por segundo y cada pulso mueve la barra 0.1mm como máximo en su régimen estable, y mientras acelera, entrega una fuerza de $(10 \pm 1)\text{N}$. Se supone que la barra de control pesa 160Kg.

Ayuda:

$$\vec{F} = m\vec{a} \rightarrow \vec{a} = \frac{\vec{F}}{m}$$

$$x = x_0 + v_0t + \frac{1}{2}at^2$$

- Analice qué sucede con el error si se considera que pueden existir *misfires* (pulsos que se envían sin que el motor reaccione).
- Determine el período en el que la barra acelerará (velocidad inicial nula). Determine, para ese período, las cotas del error en el tiempo.
- Repita el análisis del punto c) pero con la capacidad de enviar 1000 pulsos por segundo. Compare con los resultados del punto a).

5. Más errores

- Escriba dos funciones que calculen π con un número arbitrario de decimales, una en precisión simple y otra en doble precisión. Las funciones tienen que dar como resultados pi (obviamente), el número de operaciones de cada tipo usadas en el cálculo, el error numérico acumulado (absoluto, relativo y relativo porcentual) y una cota del error. Sugerencia: antes de sentarse a programar, realice el diagrama de flujo.
- Realice gráficas de cantidad de operaciones y errores en función del número de decimales. ¿Hay algún número de decimales por encima del cual no se justifique seguir agregando decimales? ¿Por qué?