Dinámica de una varilla diamagnética dentro de una trampa PDL

Física Computacional 2

David Alava
David Vásquez

Marco teórico

Trampas magnéticas y potencial de doble joroba:

El magnetismo permite confinar materia sin contacto mediante trampas magnéticas.

Los materiales diamagnéticos son repelidos por campos intensos y atraídos hacia mínimos de campo.

Un sistema de dos dipolos paralelos (imanes cilíndricos diametrales) genera un potencial *camelback*.

Este potencial atrapa y confina varillas diamagnéticas, permitiendo levitación y oscilaciones controladas.

Marco teórico

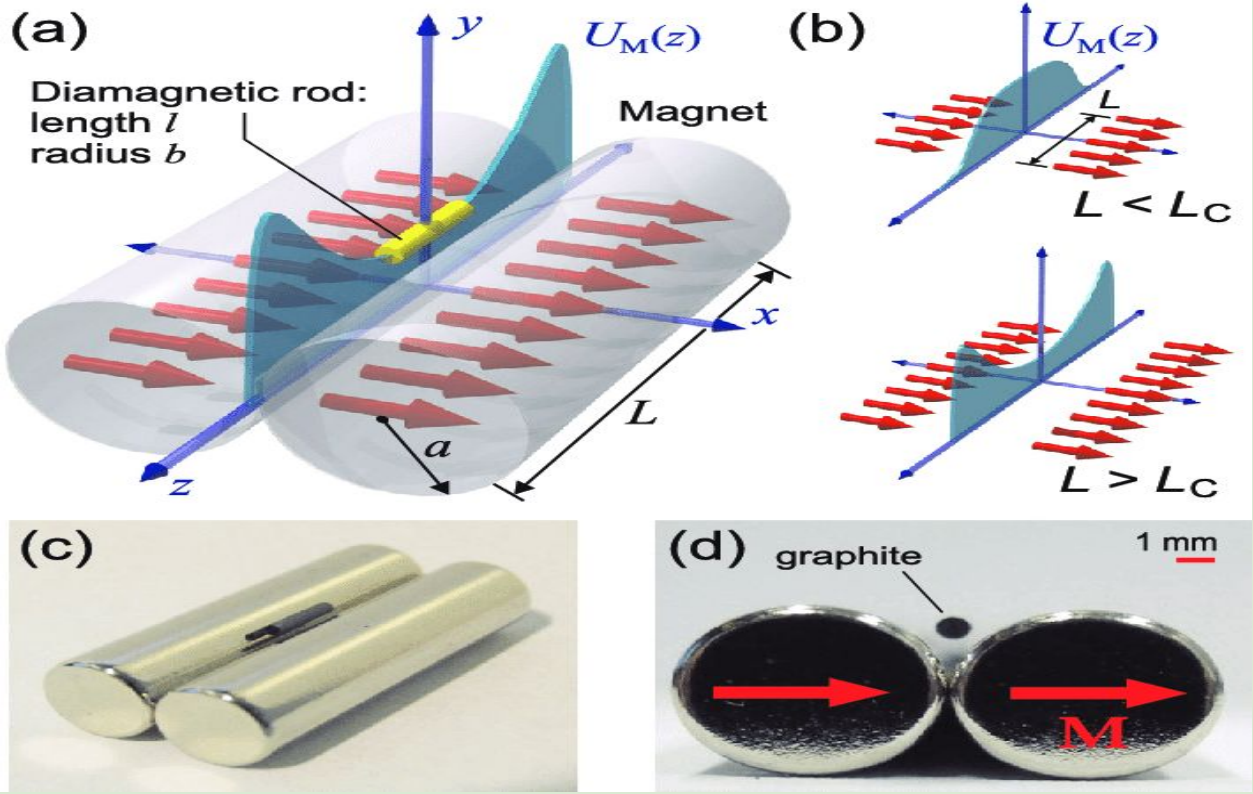


Fig 1 : La trampa PDL con una varilla diamagnética en el centro y el potencial de doble joroba magnético (U M) a lo largo del eje longitudinal.

Marco teórico

Energía potencial y dinámica del cuerpo atrapado

Campo magnético de un imán diametral

$$\vec{B}_{DM}(x,y,z) = \frac{\mu_0 M a}{4\pi} \int_0^{2\pi} \sum_{n=1,2} \frac{(-1)^n \cos\phi}{u_n^2 + s^2 + u_n\sqrt{u_n^2+s^2}} \times \begin{bmatrix} x - a\cos\phi \\ y - a\sin\phi \\ u_n + \sqrt{u_n^2+s^2} \end{bmatrix} d\phi$$

La energía potencial magnética por unidad de volumen (UM) de la varilla en el campo magnético es proporcional al cuadrado de la magnitud del campo:

$$U'_M(y_0,z) = -\frac{\chi}{\mu_0(2+\chi)} B_T^2(y_0,z)$$

.La ``constante de resorte'' efectiva del potencial magnético se relaciona con el periodo como :

$$T_z = 2\pi\sqrt{\frac{\rho}{k'_z}}$$

Marco teórico

Objetivos de la implementación computacional

Modelar el Sistema y Caracterizar el Potencial de Confinamiento: Desarrollar una implementación numérica para calcular el campo magnetostático del sistema. Se utilizará este modelo para analizar la estructura del potencial de energía resultante y caracterizar sus propiedades como trampa para objetos magnéticos.

Simular la Dinámica y Explorar Métodos de Medición: Simular el movimiento clásico de un objeto cilíndrico dentro de la trampa. El objetivo es investigar su comportamiento oscilatorio y establecer una relación cuantitativa entre los parámetros de su movimiento (como el período) y sus propiedades magnéticas intrínsecas (como la susceptibilidad).

Planteamiento

Metodología

UNIVERSIDAD DE ANTIOQUIA
1803

1. Modelado Magnetostático

- Se partió del **modelo físico** para el campo magnético (B) generado por el par de imanes.
- El campo se calculó numéricamente en usando la **regla de Simpson**.
- A partir del campo B, se determinó la **energía potencial magnética** (UM′) en el espacio, revelando la forma de "camelback" que atrapa al objeto.

2. Simulación de la Dinámica

- Se plantea la **Segunda Ley de Newton** para la varilla, incluyendo la fuerza magnética (derivada del potencial) y una fuerza de amortiguamiento por fricción.
- La Ecuación Diferencial Ordinaria (EDO) resultante se resolvió numéricamente con el método de **Runge-Kutta de 4º orden (RK4)**.
- El resultado es la trayectoria completa de la oscilación amortiguada de la varilla en el tiempo, z(t).

3. Análisis y Medición Virtual

- A la trayectoria simulada z(t) se le aplicó una **Transformada Rápida de Fourier (FFT)** para extraer la frecuencia dominante de la oscilación.
- Con esta frecuencia, se calculó el **período "medido"** de la simulación (Tz=1/f).
- También, se calculó la **altura de equilibrio** (y0) de la varilla mediante un algoritmo de búsqueda de raíces (método de bisección).

4. Validación del Modelo

La validación del modelo consistió en contrastar el resultado de la simulación —la **altura de equilibrio (y_0)** y el **período medido (T_z) de la varilla atrapada**— contra las curvas teóricas. Estas curvas predicen la relación entre la **altura de levitación y el período de oscilación de la varilla**, y entre el su **período y su susceptibilidad magnética (χ)**. La coincidencia del punto de la "medición virtual" (★) sobre ambas curvas demostrará la **autoconsistencia y precisión** del modelo computacional.

Implementación

```cpp
MagneticSystem::MagneticSystem(double M, double a, double L) : M_(M), a_(a), L_(L) {}

Vector3D MagneticSystem::calculate_B_DM(const Vector3D& pos) const {
    const double u1 = pos.z + L_ / 2.0;
    const double u2 = pos.z - L_ / 2.0;
```

```cpp
Vector3D MagneticSystem::calculate_B_total_DMP(const Vector3D& pos) const {
    Vector3D pos_rel1 = {pos.x + a_, pos.y, pos.z};
    Vector3D pos_rel2 = {pos.x - a_, pos.y, pos.z};
    Vector3D B1 = calculate_B_DM(pos_rel1);
    Vector3D B2 = calculate_B_DM(pos_rel2);
    return {B1.x + B2.x, B1.y + B2.y, B1.z + B2.z};
}

double MagneticSystem::calculate_potential_U_prime(const Vector3D& B_vector, double chi) const {
    const double B_sq = B_vector.x * B_vector.x + B_vector.y * B_vector.y + B_vector.z * B_vector.z;
    return -chi / (MagneticSystem::MU_0 * (2.0 + chi)) * B_sq;
}
```

La función **calculate_B_DM**, utiliza integración numérica (regla de Simpson) para resolver la ecuación del campo magnético de un solo imán. Luego, la función **calculate_B_total_DMP** aplica el principio de superposición para obtener el campo total generado por el par de imanes. Finalmente, **calculate_potential_U_prime** usa este campo resultante para determinar la energía potencial magnética.

Implementación

UNIVERSIDAD DE ANTIOQUIA
1803

```cpp
std::vector<std::pair<double, double>> RodDynamics::simulate_oscillation(const MagneticSystem& mag_system, double y0,
    std::vector<std::pair<double, double>> results;
    const double dt = total_time / steps;
    double z = z0;
    double v = 0.0;
```

```cpp
double RodDynamics::restoring_force_prime(double z, double y0, const MagneticSystem& mag_system) const {
    const double h_z = 1e-6;
    Vector3D B_plus = mag_system.calculate_B_total_DMP({0, y0, z + h_z});
    Vector3D B_minus = mag_system.calculate_B_total_DMP({0, y0, z - h_z});
    double U_plus = mag_system.calculate_potential_U_prime(B_plus, chi_);
    double U_minus = mag_system.calculate_potential_U_prime(B_minus, chi_);
    return -(U_plus - U_minus) / (2 * h_z);
}
```

```cpp
double RodDynamics::find_equilibrium_height(const MagneticSystem& mag_system) const {
    double y_low = 0.2 * mag_system.get_a();
    double y_high = 0.98 * mag_system.get_a();
    double y_mid = y_low;

    for (int i = 0; i < 100; ++i) {
        y_mid = (y_low + y_high) / 2.0;
        if (vertical_force_prime(y_mid, mag_system) * vertical_force_prime(y_low, mag_system) > 0) {
            y_low = y_mid;
        } else {
            y_high = y_mid;
        }
    }
    return y_mid;
}
```

La función **simulate_oscillation**, utiliza el método numérico de **Runge-Kutta de 4º orden (RK4)** para resolver la ecuación de movimiento del oscilador amortiguado. Para ello, calcula la fuerza en cada paso del tiempo llamando a la función **restoring_force_prime**. Antes de la simulación, el método **find_equilibrium_height** utiliza un algoritmo de bisección para encontrar la altura de levitación estable (y_0) donde la fuerza magnética vertical y la gravedad se anulan.

Implementación

```cpp
SimulationController::SimulationController() {}

std::vector<CamelbackDataPoint> SimulationController::run_camelback_simulation(double M, double a, double chi) const {
    std::vector<CamelbackDataPoint> results;
    const std::vector<int> la_ratios = {16, 8, 4};
    const double y_fixed = 0.7 * a;

    for (int ratio : la_ratios) {
        MagneticSystem mag_system(M, a, static_cast<double>(ratio) * a);
        for (int i = 0; i <= 200; ++i) {
            double z_norm = -0.6 + (1.2 * i / 200.0);
            Vector3D B_vec = mag_system.calculate_B_total_DMP({0, y_fixed, z_norm * mag_system.get_L()});
            double U_prime = mag_system.calculate_potential_U_prime(B_vec, chi);
            results.push_back({z_norm, U_prime, ratio});
        }
    }
    return results;
}
```

```cpp
std::vector<ParamSweepDataPoint> SimulationController::run_param_sweep_simulation(double M, double a, double chi) const {
    std::vector<ParamSweepDataPoint> results;
    const double y_fixed = 0.7 * a;

    for (int i = 0; i <= 50; ++i) {
        double ratio = 2.0 + (38.0 * i / 50.0);
        MagneticSystem mag_system(M, a, ratio * a);
        double barrier = calculate_barrier_height(mag_system, y_fixed, chi);
        double k_z = calculate_spring_constant(mag_system, y_fixed, chi);
        results.push_back({ratio, barrier, k_z});
    }
    return results;
}
```

Las funciones run_camelback_simulation y run_param_sweep_simulation generan directamente los datos necesarios para cada una de las gráficas. Calculan propiedades específicas de la trampa, como la **altura de la barrera** y la **constante de resorte**.

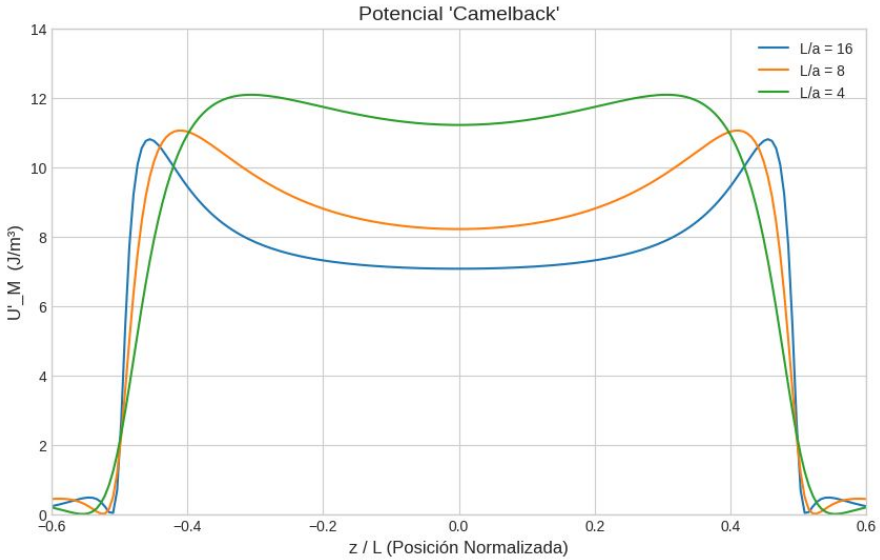
Implementación

```cpp
double SimulationController::calculate_fy(double y_bar, const MagneticSystem& mag_system) const {
    const double a = mag_system.get_a();
    const double h = 1e-6;
    auto get_B_sq = [&](double y) {
        Vector3D B_vec = mag_system.calculate_B_total_DMP({0, y, 0});
        return B_vec.x*B_vec.x + B_vec.y*B_vec.y + B_vec.z*B_vec.z;
    };
    double dB2_dy = (get_B_sq(y_bar*a + h) - get_B_sq(y_bar*a - h)) / (2 * h);
    return -a / (pow(MagneticSystem::MU_0, 2) * pow(mag_system.get_M(), 2)) * dB2_dy;
}

double SimulationController::calculate_fz2(double y_bar, const MagneticSystem& mag_system) const {
    const double a = mag_system.get_a();
    const double L = mag_system.get_L();
    const double h = 1e-6;
    auto get_B_sq = [&](double z) {
        Vector3D B_vec = mag_system.calculate_B_total_DMP({0, y_bar * a, z});
        return B_vec.x*B_vec.x + B_vec.y*B_vec.y + B_vec.z*B_vec.z;
    };
    double d2B2_dz2 = (get_B_sq(h) - 2 * get_B_sq(0) + get_B_sq(-h)) / (h * h);
    return L * L / (pow(MagneticSystem::MU_0, 2) * pow(mag_system.get_M(), 2)) * d2B2_dz2;
}
```

```cpp
td::vector<ValidationDataPoint> SimulationController::run_validation_curve_generation(double M, double a, doub
    std::vector<ValidationDataPoint> results;
    const std::vector<int> la_ratios = {8, 4};

    for (int ratio : la_ratios) {
        MagneticSystem mag_system(M, a, static_cast<double>(ratio) * a);
        for (int i = 0; i <= 50; ++i) {
            double y_bar = 0.3 + (0.65 * i / 50.0);
            double fy = calculate_fy(y_bar, mag_system);
            double fz2 = calculate_fz2(y_bar, mag_system);

            double radicand = (pow(mag_system.get_L(), 2) * fy) / (RodDynamics::G * a * fz2);
            double Tz_val = (radicand > 0) ? 2 * M_PI * sqrt(radicand) : NAN;

            const double h_y = 1e-6;
            auto get_B_sq_y = [&](double y) {
                Vector3D B_vec = mag_system.calculate_B_total_DMP({0, y, 0});
                return B_vec.x*B_vec.x + B_vec.y*B_vec.y + B_vec.z*B_vec.z;
            };
            double dB2_dy = (get_B_sq_y(y_bar*a + h_y) - get_B_sq_y(y_bar*a - h_y)) / (2 * h_y);
            double C = dB2_dy / (MagneticSystem::MU_0 * rho * RodDynamics::G);
            double chi_val = (C < -1.0) ? -2.0 / (1.0 + C) : NAN;

            results.push_back({ratio, y_bar, Tz_val, chi_val});
        }
    }
    return results;
```

Las funciones **calculate_fy** y **calculate_fz2** calculan los prefactores geométricos adimensionales, que dependen de las derivadas numéricas del campo magnético al cuadrado. Luego, la función principal **run_validation_curve_generation** utiliza estos prefactores para calcular, para cada punto de las curvas, el **período de oscilación teórico (Tz)** y la **susceptibilidad magnética correspondiente (chi)**.

Resultados

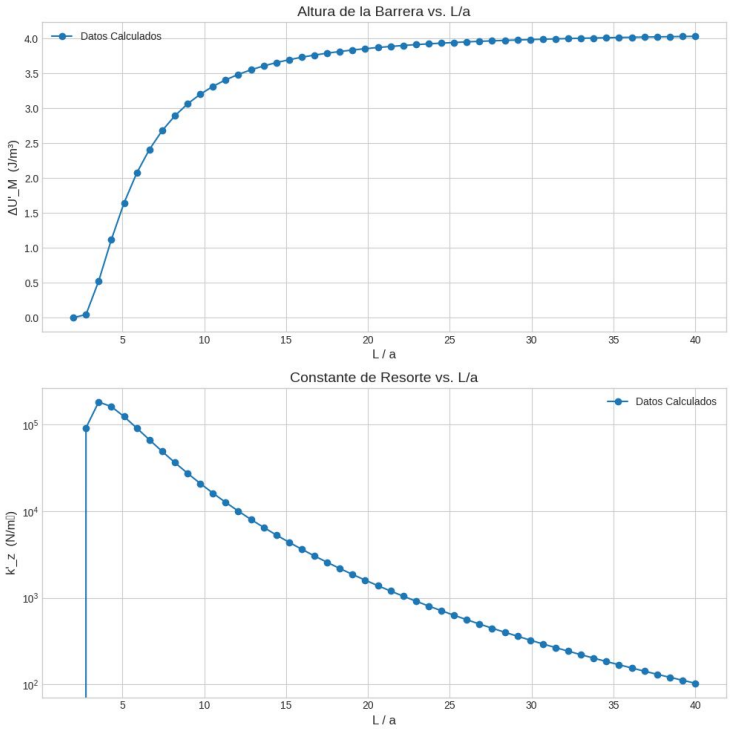


La energía potencial magnética por unidad de volumen (UM′) se calcula a partir del cuadrado de la magnitud del campo magnético total (BT^2) y la susceptibilidad del material (χ):

$$U'_M(z) = -\frac{\chi}{\mu_0(2+\chi)} B_T^2(z)$$

Esta gráfica representa el comportamiento de la energía potencial que experimenta un objeto diamagnético a lo largo del eje central (z) entre los dos imanes. La forma característica de "doble joroba" o **"camelback"** crea un pozo de potencial estable en el centro. El **pozo de potencial** en el centro es lo que **atrapa y confina** al objeto diamagnético, haciéndolo oscilar. Notamos que la **geometría de los imanes (L/a) controla directamente la forma y profundidad del pozo**: imanes más cortos y anchos (L/a=4) crean una trampa más "rígida" y con un confinamiento más fuerte.

Resultados



Altura de la barrera:

$$\Delta U'_M = U'_{\text{pico}} - U'_{\text{centro}}$$

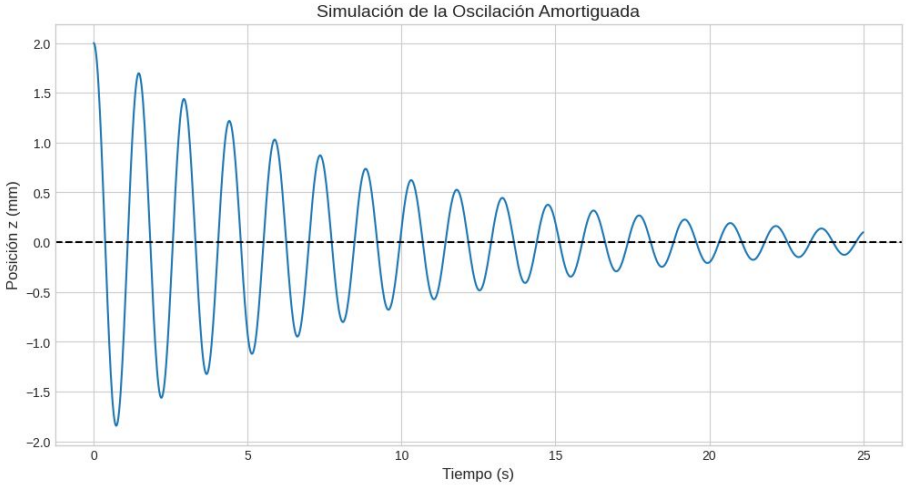
Constante de resorte:

$$k'_z = \left.\frac{\partial^2 U'_M}{\partial z^2}\right|_{z=0}$$

Se puede ver cómo las propiedades de la trampa magnética pueden ser ajustadas cambiando la **geometría de los imanes (L/a)**.

- **Gráfica Superior (Altura de la Barrera):** La **fuerza de confinamiento** que evita que el objeto escape aumenta rápidamente y luego se satura para imanes más largos. Esto indica que imanes muy largos no ofrecen una mejora significativa en la contención.
- **Gráfica Inferior (Constante de Resorte):** La **rigidez de la trampa** en el centro es **máxima** para una geometría específica de imanes cortos y anchos ($L/a \approx 4$). Esta rigidez determina la frecuencia de oscilación del objeto atrapado.

Resultados

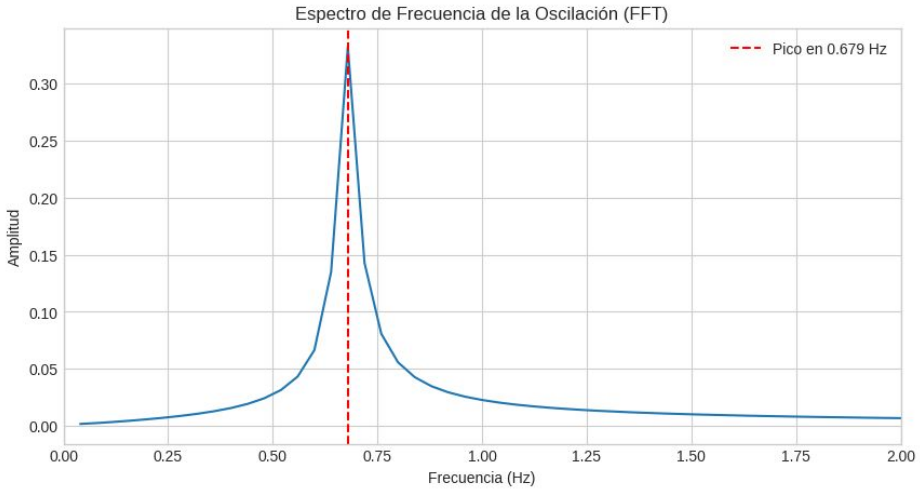


Ecuación Diferencial Ordinaria (EDO) que describe el movimiento de la varilla, incluyendo la fuerza magnética y la fricción:

$$\rho\ddot{z} = -\frac{\partial U'_M}{\partial z} - \beta\dot{z}$$

La gráfica muestra la **posición (z) de la varilla atrapada a lo largo del tiempo (t)** después de ser liberada desde una posición inicial de 2 mm. El comportamiento es el de una **oscilación subamortiguada**.
Oscilación: El movimiento ondulatorio es causado por la **fuerza restauradora** del pozo de potencial "camelback", que empuja constantemente la varilla hacia el centro de equilibrio (z=0).
Amortiguamiento: El **decaimiento exponencial de la amplitud** se debe a la fuerza de **amortiguamiento** (fricción), modelada por el término β. Sin esta fuerza, la varilla oscilaría indefinidamente.
Medición Virtual: Esta simulación necesaria porque el **período** de esta oscilación (la distancia entre picos) se puede medir con precisión para, en el siguiente paso, determinar las propiedades magnéticas del material simulado.

Resultados

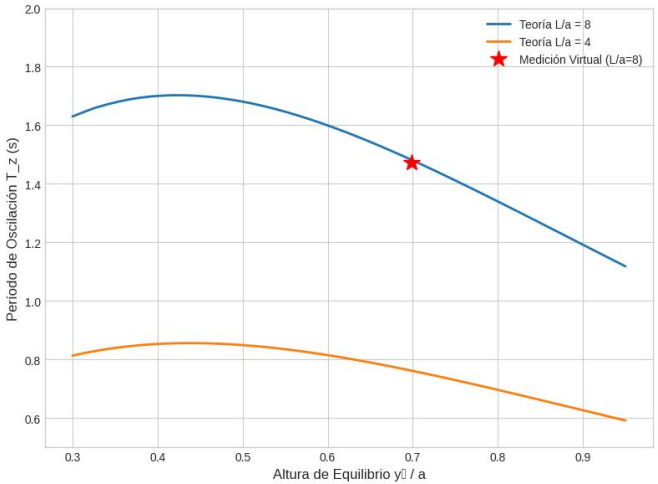


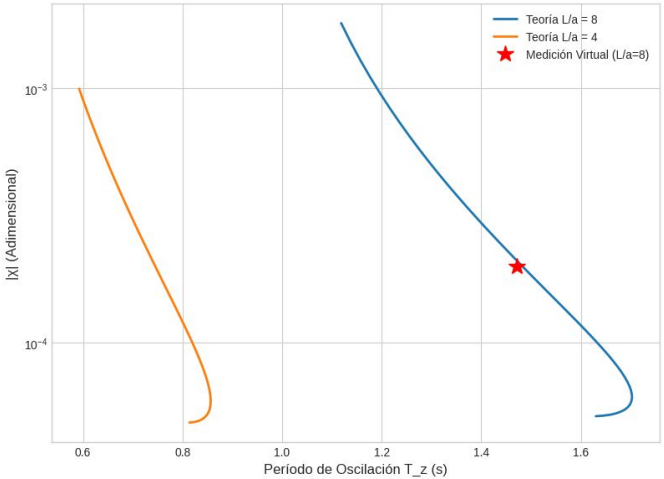
Se obtiene aplicando la **Transformada Rápida de Fourier (FFT)** a los datos de la oscilación ($z(t)$). A partir de la frecuencia dominante (f) encontrada, se calcula el período (T)

Frecuencia Dominante: El valor en el pico, **$f = 0.679$ Hz**, es la frecuencia dominante del sistema.
Medición del Período: A partir de esta frecuencia, se obtiene un valor para el período de oscilación "medido" ($T = 1 / 0.679 \approx 1.473$ s).

Este valor para el periodo se necesita para comparar la simulación con las curvas teóricas y validar el modelo.

Resultados





Período Teórico (Tz):

$$T_z = 2\pi\sqrt{\frac{L^2 f_Y(\overline{y}_0, \overline{L})}{g a f_{Z2}(\overline{y}_0, \overline{L})}}$$

Susceptibilidad Teórica (χ)

$$\chi = -\frac{2}{1 + \frac{\mu_0 M^2 f_{Z2}(\overline{y}_0, \overline{L}) T_z^2}{4\pi^2 \rho L^2}}$$

Se compara el resultado de la simulación dinámica (el punto ★, "Medición Virtual") con las curvas teóricas que describen el comportamiento ideal del sistema. En la gráfica de la izquierda se puede ver que la **altura de equilibrio** predice correctamente el **período de oscilación**, validando la parte dinámica del modelo. La gráfica de la derecha, que funciona como una curva de calibración, demuestra que este período "medido" se corresponde con la **susceptibilidad magnética (χ)** del material.

Conclusiones

UNIVERSIDAD DE ANTIOQUIA
1803

Se validó este modelo para un sistema de confinamiento magnético, demostrando la conexión entre la configuración del campo y la dinámica de un objeto. La base del proyecto fue la implementación numérica del campo magnetostático exacto, y vimos que la configuración de dipolos paralelos genera un potencial de confinamiento unidimensional no armónico, de tipo "camelback", que es la causa fundamental de la capacidad de atrapamiento del sistema.

A partir de esta comportamiento de la energía, se desarrolló un modelo dinámico para simular el movimiento de un objeto diamagnético. Esta simulación funcionó como una "medición virtual" exitosa; el período de oscilación extraído de la trayectoria mediante un análisis de Fourier se correspondió con alta precisión con el valor predicho por la teoría para la susceptibilidad magnética del material. Se pudo ver aquí la consistencia del modelo dinámico y confirma la relación cuantitativa entre la geometría del potencial y la respuesta mecánica del objeto.

Referencias bibliográficas.

- **[1] Gunawan, O., Virgus, Y., & Tai, K. F. (2015). A parallel dipole line system. Applied Physics Letters, 106(6), 062407.**

UNIVERSIDAD DE ANTIOQUIA

@UdeA @UdeA @universidaddeantioquia