



Corporación Universitaria Americana

Facultad de Ingeniería

Sistema Masa-Resorte Interactivo

Ecuaciones diferenciales

Elaborado por:

Carlos Arturo López Manga
Ricardo David Ruiz Pertuz
Keiner Javier Ortega Blanco
Jonathan Javer Barrios Uribe
Andrés Felipe Escorcía Villanueva

Asignatura: Ecuaciones Diferenciales
Docente: Luis Bedoya

Barranquilla, Colombia
24 de noviembre de 2025

Índice

1. Introducción y Motivación	4
1.1. El Problema de la Resonancia en la Ingeniería	4
1.2. Propósito del Proyecto	4
1.3. Relevancia Educativa	4
2. Marco Teórico	4
2.1. La Ecuación Fundamental	4
2.2. Interpretación Física de los Términos	5
2.2.1. Término de Inercia: my''	5
2.2.2. Término de Amortiguamiento: cy'	5
2.2.3. Término de Restauración: ky	5
2.3. Frecuencia Natural y Resonancia	5
2.4. Tipos de Amortiguamiento	6
3. Diseño e Implementación	6
3.1. Arquitectura del Software	6
3.2. Estructura del Código	6
3.2.1. Clase WelcomeScreen	6
3.2.2. Clase MassSpringApp	6
3.3. Algoritmo de Solución Numérica	7
3.4. Implementación de la Ecuación Diferencial	7
4. Características del Simulador	7
4.1. Parámetros Ajustables	7
4.2. Tipos de Fuerza Externa	8
4.3. Experimentos Predefinidos	8

4.3.1. Experimento 'Normal'	8
4.3.2. Experimento 'Resonancia'	8
4.3.3. Experimento 'Amortiguado'	8
4.3.4. Experimento 'Libre'	8
4.4. Visualizaciones	9
5. Análisis de Resultados	9
5.1. Verificación del Modelo	9
5.1.1. Caso Sin Amortiguamiento ni Fuerza Externa	9
5.1.2. Resonancia	9
5.2. Ejemplos de Comportamiento Observado	9
5.2.1. Resonancia con Bajo Amortiguamiento	9
5.2.2. Amortiguamiento Crítico	9
6. Aplicaciones en el Mundo Real	10
6.1. Ingeniería Civil	10
6.1.1. Edificios Altos	10
6.1.2. Puentes	10
6.2. Ingeniería Automotriz	10
6.3. Instrumentos Musicales	10
6.4. Electrónica	10
7. Metodología de Desarrollo	11
7.1. Enfoque Iterativo	11
7.2. Pruebas y Validación	11
8. Retos y Soluciones	11
8.1. Retos Técnicos	11
8.1.1. Estabilidad Numérica	11

8.1.2. Rendimiento en Tiempo Real	11
8.1.3. Precisión Visual	12
8.2. Retos Educativos	12
8.2.1. Abstracción Matemática	12
8.2.2. Conceptos Contra-intuitivos	12
9. Contribuciones y Originalidad	12
9.1. Aportes Educativos	12
9.2. Innovaciones Técnicas	12
10.Conclusiones y Trabajo Futuro	13
10.1. Logros Principales	13
10.2. Limitaciones Actuales	13
10.3. Direcciones Futuras	13
10.3.1. Mejoras al Simulador	13
10.3.2. Extensiones Educativas	13
10.3.3. Investigación	14
11.Impacto y Aplicaciones	14
11.1. Impacto Educativo	14
11.2. Aplicaciones Prácticas	14

1 Introducción y Motivación

1.1 El Problema de la Resonancia en la Ingeniería

La resonancia mecánica es un fenómeno físico que ha causado algunos de los desastres de ingeniería más famosos de la historia. El caso más emblemático es el colapso del Puente de Tacoma Narrows en 1940, donde vientos moderados de apenas 68 km/h generaron oscilaciones que destruyeron la estructura en pocas horas. Este evento demostró dramáticamente la importancia de entender los sistemas oscilatorios y sus frecuencias naturales.

1.2 Propósito del Proyecto

Este proyecto busca crear una herramienta educativa interactiva que permita visualizar y experimentar con los conceptos fundamentales de los sistemas masa-resorte. El objetivo principal es hacer accesibles conceptos matemáticos complejos (ecuaciones diferenciales, métodos numéricos) a través de una interfaz visual e interactiva.

1.3 Relevancia Educativa

El sistema masa-resorte es fundamental en la física y la ingeniería porque:

- Es el modelo más simple de oscilador armónico
- Aparece en múltiples contextos (mecánica, electricidad, acústica)
- Introduce conceptos clave como frecuencia natural y resonancia
- Sirve como base para sistemas más complejos

2 Marco Teórico

2.1 La Ecuación Fundamental

El comportamiento del sistema masa-resorte está gobernado por la ecuación diferencial:

$$m \frac{d^2 y}{dt^2} + c \frac{dy}{dt} + ky = F(t) \quad (1)$$

Donde:

- m : masa (kg)

- c : coeficiente de amortiguamiento (N·s/m)
- k : constante elástica del resorte (N/m)
- $F(t)$: fuerza externa aplicada (N)
- y : desplazamiento desde la posición de equilibrio (m)

2.2 Interpretación Física de los Términos

2.2.1. Término de Inercia: my''

Representa la resistencia de la masa a cambiar su estado de movimiento. Según la segunda ley de Newton, $F = ma$, donde la aceleración es la segunda derivada de la posición.

2.2.2. Término de Amortiguamiento: cy'

Modela la fricción o resistencia que disipa energía del sistema. Es proporcional a la velocidad, lo que significa que a mayor velocidad, mayor fuerza de amortiguamiento.

2.2.3. Término de Restauración: ky

Representa la fuerza elástica del resorte que intenta devolver el sistema a su posición de equilibrio. Sigue la ley de Hooke: $F = -ky$.

2.3 Frecuencia Natural y Resonancia

La frecuencia natural del sistema está dada por:

$$\omega_n = \sqrt{\frac{k}{m}} \quad (2)$$

La resonancia ocurre cuando la frecuencia de la fuerza externa coincide con la frecuencia natural:

$$\omega_{ext} = \omega_n \quad (3)$$

En este punto, pequeñas fuerzas externas pueden producir amplitudes de oscilación muy grandes.

2.4 Tipos de Amortiguamiento

- **Subamortiguado** ($c < 2\sqrt{mk}$): El sistema oscila mientras la amplitud decae exponencialmente
- **Críticamente amortiguado** ($c = 2\sqrt{mk}$): El sistema retorna al equilibrio en el menor tiempo posible sin oscilar
- **Sobreamortiguado** ($c > 2\sqrt{mk}$): El sistema retorna al equilibrio lentamente sin oscilar

3 Diseño e Implementación

3.1 Arquitectura del Software

El simulador fue desarrollado en Python utilizando las siguientes bibliotecas:

- **NumPy**: Para cálculos numéricos y operaciones con arrays
- **SciPy**: Para resolver ecuaciones diferenciales con `solve_ivp`
- **Matplotlib**: Para visualización y animaciones
- **Tkinter**: Para la interfaz gráfica de usuario

3.2 Estructura del Código

El programa está organizado en dos clases principales:

3.2.1. Clase WelcomeScreen

Gestiona la pantalla de bienvenida con explicaciones iniciales y el botón para iniciar la simulación.

3.2.2. Clase MassSpringApp

Contiene la lógica principal de la aplicación:

- Interfaz de usuario con controles deslizantes
- Sistema de animación en tiempo real

- Resolución de ecuaciones diferenciales
- Visualización de resultados

3.3 Algoritmo de Solución Numérica

Se utiliza el método Runge-Kutta de orden 4/5 adaptativo (RK45) para resolver la ecuación diferencial:

```
1 def solve_system(self):
2     t_eval = np.linspace(0, 20, 800)
3     sol = solve_ivp(
4         self.equation,
5         [0, 20],
6         [0, 0],
7         t_eval=t_eval,
8         method='RK45'
9     )
10    return sol.t, sol.y[0]
```

Listing 1: Método de solución numérica

3.4 Implementación de la Ecuación Diferencial

La ecuación de segundo orden se transforma en un sistema de dos ecuaciones de primer orden:

```
1 def equation(self, t, Y):
2     y, yp = Y # posicion y velocidad
3     force = self.external_force(t)
4
5     # Sistema de primer orden
6     dydt = yp # dy/dt = velocidad
7     dypdt = (-self.k * y - self.c * yp + force) / self.m
8
9     return [dydt, dypdt]
```

Listing 2: Implementación del sistema de ecuaciones

4 Características del Simulador

4.1 Parámetros Ajustables

Los usuarios pueden modificar en tiempo real:

- Masa (m): 0.1 a 5.0 kg

- **Rigidez (k):** 0.5 a 15.0 N/m
- **Amortiguamiento (c):** 0.0 a 2.0 N·s/m
- **Fuerza externa (F):** 0 a 10 N
- **Frecuencia externa (ω):** 0.1 a 8.0 rad/s

4.2 Tipos de Fuerza Externa

El simulador implementa cuatro tipos de fuerza:

- **Coseno:** $F(t) = F_0 \cos(\omega t)$
- **Seno:** $F(t) = F_0 \sin(\omega t)$
- **Pulso:** Fuerza que alterna entre 0 y F_0
- **Escalón:** Fuerza constante que se activa después de 2 segundos

4.3 Experimentos Predefinidos

Se incluyen cuatro configuraciones predefinidas:

4.3.1. Experimento 'Normal'

Configuración balanceada que muestra oscilaciones regulares.

4.3.2. Experimento 'Resonancia'

Parámetros optimizados para demostrar resonancia dramática.

4.3.3. Experimento 'Amortiguado'

Muestra el efecto de alto amortiguamiento en las oscilaciones.

4.3.4. Experimento 'Libre'

Sistema sin fuerzas externas ni amortiguamiento.

4.4 Visualizaciones

El simulador proporciona múltiples formas de visualizar los resultados:

- Animación en tiempo real del sistema masa-resorte
- Gráfica de posición vs tiempo
- Indicadores de resonancia con cambio de colores
- Panel de información con parámetros actuales

5 Análisis de Resultados

5.1 Verificación del Modelo

Para validar el simulador, se compararon los resultados con soluciones analíticas conocidas:

5.1.1. Caso Sin Amortiguamiento ni Fuerza Externa

La solución analítica es:

$$y(t) = A \cos(\omega_n t + \phi) \quad (4)$$

El simulador reproduce exactamente este comportamiento.

5.1.2. Resonancia

Cuando $\omega_{ext} = \omega_n$, el simulador muestra el crecimiento esperado en la amplitud de oscilación.

5.2 Ejemplos de Comportamiento Observado

5.2.1. Resonancia con Bajo Amortiguamiento

Con $m = 1,0$ kg, $k = 4,0$ N/m, $c = 0,05$ N·s/m, y $F_0 = 3,0$ N a $\omega = 2,0$ rad/s, se observa resonancia dramática con amplitudes que crecen continuamente.

5.2.2. Amortiguamiento Crítico

Con $m = 2,0$ kg, $k = 4,0$ N/m, y $c \approx 5,66$ N·s/m, el sistema retorna al equilibrio en el menor tiempo posible sin oscilar.

6 Aplicaciones en el Mundo Real

6.1 Ingeniería Civil

6.1.1. Edificios Altos

El Taipei 101 utiliza un amortiguador de masa sintonizada de 660 toneladas para reducir oscilaciones causadas por vientos y terremotos.

6.1.2. Puentes

El diseño moderno de puentes considera cuidadosamente las frecuencias naturales para evitar resonancia con el viento o el tráfico.

6.2 Ingeniería Automotriz

Los sistemas de suspensión de vehículos son esencialmente sistemas masa-resorte-amortiguador optimizados para confort y seguridad.

6.3 Instrumentos Musicales

La caja de resonancia de guitarras y violines amplifica las vibraciones de las cuerdas mediante principios de resonancia.

6.4 Electrónica

Circuitos RLC siguen ecuaciones análogas a los sistemas masa-resorte, donde:

- Inductancia (L) Masa (m)
- Resistencia (R) Amortiguamiento (c)
- Capacitancia (C) $1/\text{Rigidez}$ ($1/k$)

7 Metodología de Desarrollo

7.1 Enfoque Iterativo

El desarrollo siguió una metodología iterativa:

1. **Prototipo inicial:** Implementación básica de la ecuación diferencial
2. **Interfaz de usuario:** Adición de controles interactivos
3. **Visualización:** Implementación de animaciones y gráficas
4. **Optimización:** Mejora del rendimiento y usabilidad
5. **Documentación:** Creación de guías y materiales educativos

7.2 Pruebas y Validación

Cada iteración incluyó pruebas exhaustivas:

- Verificación de soluciones contra casos analíticos conocidos
- Pruebas de estabilidad numérica con diferentes parámetros
- Validación de la interfaz de usuario con usuarios reales
- Pruebas de rendimiento con simulaciones largas

8 Retos y Soluciones

8.1 Retos Técnicos

8.1.1. Estabilidad Numérica

El uso de métodos numéricos para ecuaciones diferenciales puede introducir inestabilidades. Se solucionó utilizando el método RK45 adaptativo de SciPy.

8.1.2. Rendimiento en Tiempo Real

La animación en tiempo real requería optimizar los cálculos. Se implementó pre-cálculo de la solución y actualización eficiente de gráficas.

8.1.3. Precisión Visual

La representación visual del resorte necesitaba balancear realismo y claridad. Se implementó una función que genera coordenadas de resorte basadas en sinusoides.

8.2 Retos Educativos

8.2.1. Abstracción Matemática

Las ecuaciones diferenciales son abstractas para muchos estudiantes. La visualización en tiempo real ayuda a conectar matemáticas con fenómenos físicos observables.

8.2.2. Conceptos Contra-intuitivos

La resonancia puede ser contra-intuitiva (fuerzas pequeñas producen efectos grandes). Los experimentos predefinidos guían a los usuarios hacia este descubrimiento.

9 Contribuciones y Originalidad

9.1 Aportes Educativos

- **Accesibilidad:** Hace conceptos avanzados accesibles a estudiantes de secundaria
- **Interactividad:** Permite experimentación activa en lugar de observación pasiva
- **Contextualización:** Conecta teoría abstracta con aplicaciones del mundo real
- **Multidisciplinariedad:** Integra física, matemáticas y programación

9.2 Innovaciones Técnicas

- Implementación eficiente de métodos numéricos en Python
- Sistema de detección automática de resonancia
- Interfaz intuitiva que oculta complejidad técnica
- Visualizaciones que muestran múltiples aspectos del sistema simultáneamente

10 Conclusiones y Trabajo Futuro

10.1 Logros Principales

El proyecto ha demostrado exitosamente que es posible:

- Crear una herramienta educativa efectiva para enseñar sistemas dinámicos
- Implementar métodos numéricos robustos para ecuaciones diferenciales
- Diseñar interfaces intuitivas para conceptos matemáticos complejos
- Conectar teoría abstracta con aplicaciones prácticas

10.2 Limitaciones Actuales

- El modelo es lineal y no captura comportamientos no lineales
- Solo considera movimiento en una dimensión
- No incluye efectos como rozamiento seco o límites físicos
- La visualización tiene limitaciones para frecuencias muy altas

10.3 Direcciones Futuras

10.3.1. Mejoras al Simulador

- Implementación de sistemas no lineales
- Adición de múltiples grados de libertad
- Visualización del espacio de fases
- Análisis de Fourier en tiempo real
- Exportación de datos para análisis externo

10.3.2. Extensiones Educativas

- Creación de planes de lecciones estructurados
- Desarrollo de versiones para diferentes niveles educativos
- Integración con plataformas de aprendizaje en línea
- Versiones en otros idiomas

10.3.3. Investigación

- Estudio de efectividad educativa del simulador
- Comparación con otros métodos de enseñanza
- Adaptación para necesidades especiales educativas

11 Impacto y Aplicaciones

11.1 Impacto Educativo

El simulador tiene potencial para transformar la enseñanza de:

- Física de oscilaciones y ondas
- Ecuaciones diferenciales
- Métodos numéricos
- Sistemas dinámicos

11.2 Aplicaciones Prácticas

Más allá del aula, el simulador puede ser útil para:

- Ingenieros que diseñan sistemas vibratorios
- Estudiantes que preparan olimpiadas de física
- Profesores que buscan demostraciones interactivas
- Entusiastas que quieren entender fenómenos físicos

Apéndice A: Código Fuente

El código fuente completo está disponible en:

[\[https://github.com/carlop10/sistema-masa-resorte-interactivo\]](https://github.com/carlop10/sistema-masa-resorte-interactivo)

Apéndice B: Instrucciones de Uso

Prerrequisitos

- Python 3.8 o superior
- pip (gestor de paquetes de Python)

Ejecución

Instalación en 4 pasos

1. Clonar el repositorio:

```
git clone https://github.com/carlop10/sistema-masa-resorte-interactivo.git
cd sistema-masa-resorte-interactivo
```

2. Crear y activar el entorno virtual:

```
python -m venv venv
venv\Scripts\activate      (Windows)
source venv/bin/activate   (Linux/Mac)
```

3. Instalar dependencias:

```
pip install -r requirements.txt
```

4. Ejecutar la aplicación:

```
python main.py
```