KONRAD LORENZ
FUNDACIÓN UNIVERSITARIA

Acreditación Institucional de **Alta Calidad**

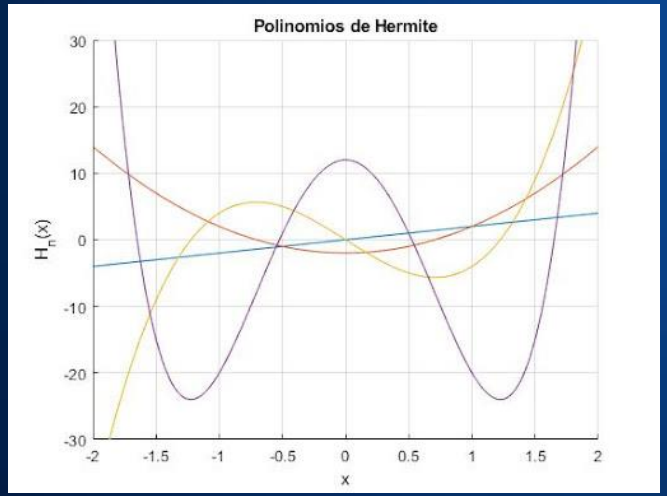
Automatización del Método de Turbiner para Ecuaciones Diferenciales con Soluciones Polinomiales

Nicolás Moreno Martínez
Esteban Felipe Tellez Ruiz

Resumen del artículo

¿Cómo identificar sistemáticamente ecuaciones diferenciales que tienen soluciones polinomiales exactas?

Por ejemplo, el oscilador armónico cuántico tiene como soluciones los polinomios de Hermite. Pero, ¿cómo sabemos qué otras ecuaciones tienen esta propiedad?



$$(n+2)(n+1)c_{n+2} - 2\alpha n c_n + \left(\frac{2mE}{\hbar^2} - \alpha\right)c_n = 0$$

Reescribiendo la ecuación que relacione C_n y C_{n+2} :

$$(n+2)(n+1)c_{n+2} - 2\alpha n c_n + \left(\frac{2mE}{\hbar^2}c_n - \alpha c_n\right) = 0$$

$$(n+2)(n+1)c_{n+2} - 2\alpha n c_n + \frac{2mE}{\hbar^2}C_n - \alpha C_n = 0$$

Despejando :

$$(n+2)(n+1)c_{n+2} = \alpha C_n + 2\alpha n c_n - \frac{2mE}{\hbar^2}C_n$$

Factorizando C_n :

$$(n+2)(n+1)c_{n+2} = Cn\left[\ \alpha + 2\alpha n - \frac{2mE}{\hbar^2}\ \right]$$

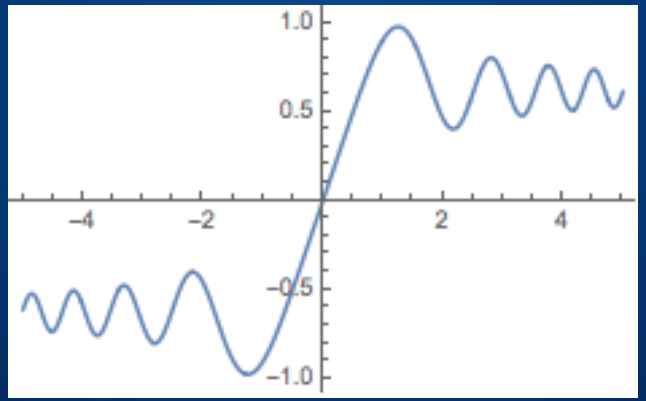
Despejando C_{n+2} :

$$C_{n+2} = \frac{\left[\ \alpha + 2\alpha n - \frac{2mE}{\hbar^2}\ \right]}{(n+2)(n+1)} \cdot Cn$$

EL PROBLEMA GENERALIZADO DE BOCHNER

¿Históricamente, Bochner preguntó qué ecuaciones tienen INFINITAS soluciones polinomiales ortogonales. Esto ya había sido resuelto por Krall en 1938.

Turbiner generalizó esto: ¿Qué ecuaciones tienen un NÚMERO FINITO de soluciones polinomiales? Este problema era completamente nuevo.



LA SOLUCIÓN - ÁLGEBRA DE LIE

La solución de Turbiner es elegante: demostró que **todas** las ecuaciones con soluciones polinomiales se pueden construir usando solo 3 operadores del álgebra de Lie $sl_2(R)$:

$J_+ = x^2\partial_x - nx$ (operador de subida)
$J_0 = x\partial_x - n/2$ (operador neutro)
$J_- = \partial_x$ (operador de bajada)

Donde n define el espacio de polinomios $P_n = \{1, x, x^2, ..., x^n\}$"

CONCEPTO CLAVE - PRESERVACIÓN

El concepto central es la PRESERVACIÓN de espacios. Un operador preserva P_n si transforma cualquier polinomio de grado $\leq n$ en otro polinomio de grado $\leq n$.

Por ejemplo, si aplicamos un operador a x^2 y obtenemos $3x^3$, eso está en P_3. Pero si obtenemos x^5, este no está en P_3.

$$P(x) = 2x^3 + 3x^2 + 5x + 8$$

TEOREMA

El teorema principal de Turbiner establece:

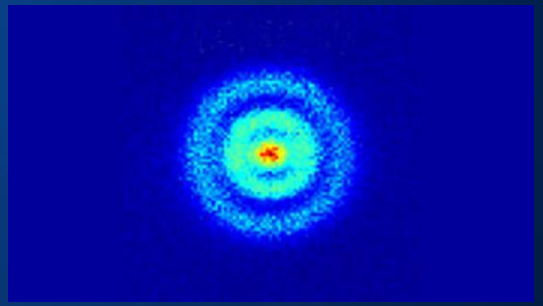
Una ecuación diferencial $T \cdot \varphi = \lambda \cdot \varphi$ tiene exactamente (n+1) soluciones polinomiales si y solo si el operador T puede escribirse como combinación polinomial de J_+, J_0, J_-.

Esto resuelve completamente el problema generalizado de Bochner.

IMPACTO Y APLICACIONES

Este resultado unifica toda la teoría de polinomios ortogonales clásicos y proporciona el marco teórico para los problemas cuasi-exactamente resolubles en mecánica cuántica, donde podemos calcular exactamente algunos estados de energía.

- Ejemplos: oscilador armónico, átomo de hidrógeno
- Conexión con física cuántica



IMPLEMENTACIÓN COMPUTACIONAL - APORTE

Justificación y Arquitectura del Software

A pesar de la elegancia de la teoría de Turbiner, no existía ninguna herramienta computacional que la implementara. Todo el trabajo se hacía manualmente, ecuación por ecuación.

El proyecto llena este vacío.

Para ello, se desarrolló una herramienta en Python con 3 módulos principales:

- **Módulo de Generadores:** Implementa J_+, J_0, J_- usando SymPy.
- **Módulo de Construcción/ Solución :** Permite crear operadores QES personalizados y usa álgebra lineal para resolver eigenvalores.
- **Módulo de Visualización:** Grafica las soluciones.

ENFOQUE MATRICIAL

La estrategia clave fue representar el operador T como una matriz en la base $\{1, x, x^2, ..., x^n\}$.

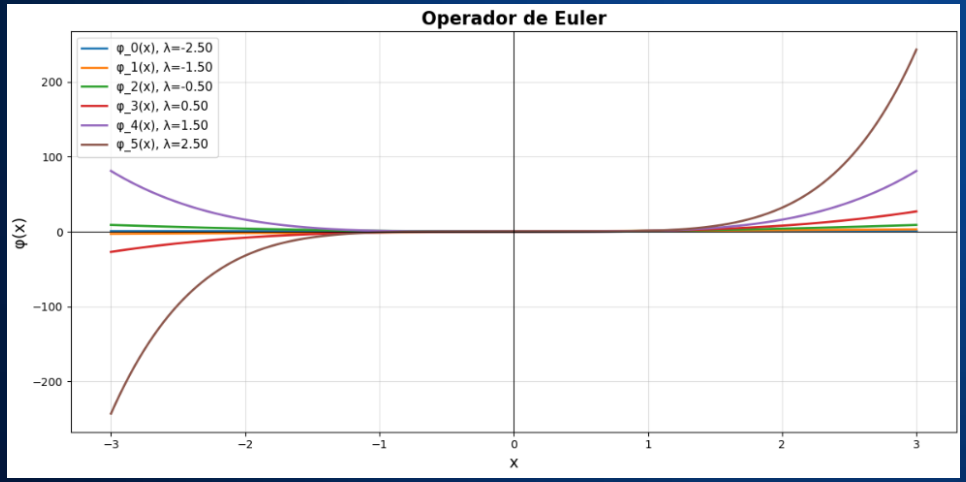
Por ejemplo, para n=2:

- Aplicamos T a cada elemento: $T(1)$, $T(x)$, $T(x^2)$.
- Extraemos coeficientes y construimos la matriz M.
- Los eigenvalores de M son los λ de la ecuación $T \cdot \varphi = \lambda \cdot \varphi$
- Los eigenvectores nos dan los coeficientes de los polinomios solución

CÓDIGO EJEMPLO

El uso es muy sencillo. Para crear el operador de Euler

```
operador = {(0,1,0): 1} # Solo j0
sols_euler = resolver(operador, n=5)
graficar(sols_euler, "Operador de Euler")
```



El operador de Euler en una variable es:

$$E = x\frac{d}{dx}$$

En la representación usada aparece como:

$$J_0 = x\frac{d}{dx} - \frac{n}{2}$$

Las soluciones son potencias:

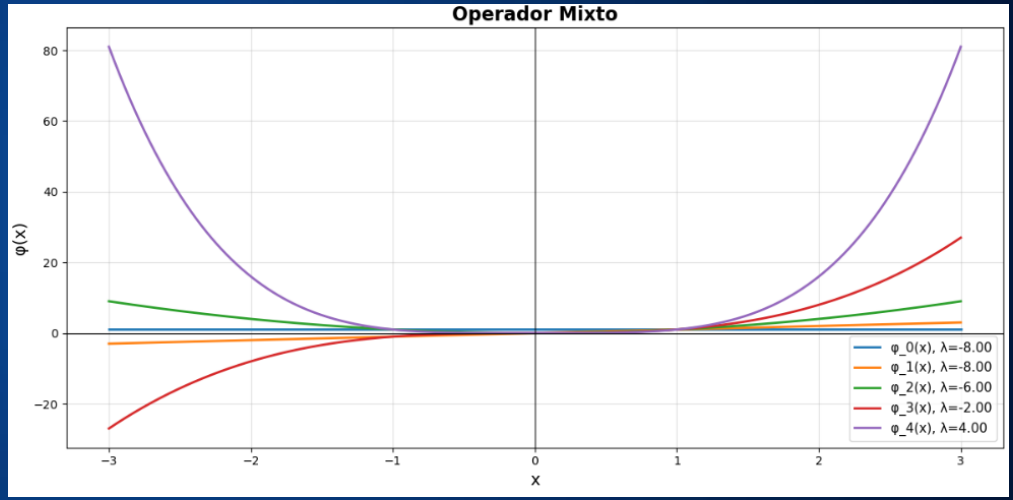
$$\ell(x) = C|x|^{\lambda+n/2}$$

El operador definido en el código reproduce correctamente el comportamiento teórico del generador J_0 del álgebra sl2. Las soluciones numéricas coinciden con la solución analítica esperada.

CÓDIGO EJEMPLO

Otro ejemplo pero con un operador mixto:

```
# Operador: T = J₊ · J₋ + 2J₀
operador_mixto = {
    (1,0,1): 1,    # J₊·J₋
    (0,1,0): 2     # 2J₀
}
sols = resolver(operador_mixto, n=4)
graficar(sols, "Operador Mixto")
```



Todos los resultados confirman que los operadores construidos con los generadores sí preservan espacios polinomiales.

LIMITACIONES ENCONTRADAS

También se encontró limitaciones:

- Se intentó reproducir el oscilador armónico directamente pero requiere transformaciones más complejas.

- Para grados muy altos ($n > 20$), aparecen problemas de precisión numérica.

- No todos los operadores físicos importantes son directamente expresables en esta forma.

Esto confirma que la teoría de Turbiner, aunque completa, requiere trabajo adicional para conectar con ciertos problemas físicos."

Conclusiones

Logros del proyecto:

Primera implementación computacional del método de Turbiner, validación exitosa con casos conocidos, código abierto y documentado. Con oportunidades de mejoras e implementación más rigurosa.

$$J_+ = x^2\partial_x - nx$$
$$J_0 = x\partial_x - n/2$$
$$J_- = \partial_x$$

KONRAD LORENZ
FUNDACIÓN UNIVERSITARIA

Acreditación Institucional de **Alta Calidad**

¡Gracias!