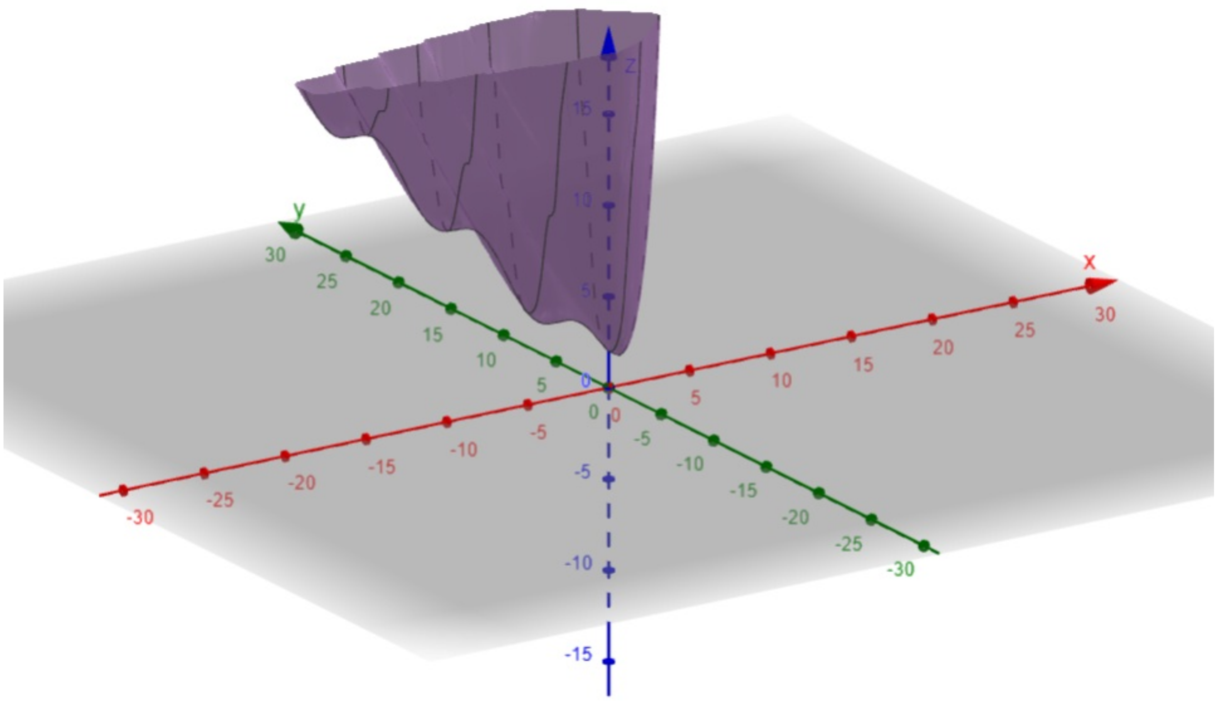
Universidad de La Habana

Facultad de Matemática y Computación

TAREA EVALUATIVA

Asignatura: Modelos de Optimización

Nombre: Fabio Víctor Alonso Bañobre
Grupo: C-311



Modelo a Analizar

$$\min f(x,y) = (e^x + 1)(y^2 + 1) - \sin(x + y^2) - x$$

Análisis Teórico del Modelo

Estamos en presencia de un problema de optimización no lineal irrestricto.

Existencia de Solución

Condición para existencia de solución:

$$\lim_{\|(x,y)\|\to\infty} f(x,y) = +\infty$$

Además, la función $f(x,y)$ es continua en todo $\mathbb{R}^2$ por ser composición de funciones continuas (exponencial, polinómica, trigonométrica y sus combinaciones).

En nuestro caso

Analizar el cumplimiento de:

$$\lim_{\|(x,y)\|\to\infty} \left[(e^x + 1)(y^2 + 1) - \sin(x + y^2) - x\right] = +\infty$$

Para verificar este límite, consideramos los límites por separado. Si cuando $x \to \pm\infty$ o $y \to \pm\infty$ la función tiende a $+\infty$, entonces el límite conjunto también tiende a $+\infty$. En efecto:

- Cuando $x \to +\infty$, e^x domina $\Rightarrow f(x,y) \to +\infty$
- Cuando $x \to -\infty$, $-x$ domina $\Rightarrow f(x,y) \to +\infty$
- Cuando $y \to \pm\infty$, y^2 domina $\Rightarrow f(x,y) \to +\infty$

Por lo tanto, la función garantiza la existencia de al menos un mínimo global

Convexidad

Condición de convexidad para una función $f \in C^2$: La matriz Hessiana $\nabla^2 f(x,y)$ debe ser semidefinida positiva $\forall (x,y) \in \mathbb{R}^2$.

El gradiente de la función es:

$$\nabla f(x,y) = \big(e^x(y^2+1) - \cos(x+y^2) - 1,\ 2(e^x+1)y - 2y\cos(x+y^2)\big)$$

La matriz Hessiana tiene la forma:

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}$$

Valores de las derivadas:

$$\frac{\partial^2 f}{\partial x^2} = e^x(y^2+1) + \sin(x+y^2)$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} = 2ye^x + 2y\sin(x+y^2)$$

$$\frac{\partial^2 f}{\partial y^2} = 2(e^x+1) - 2\cos(x+y^2) + 4y^2\sin(x+y^2)$$

Por lo tanto, la matriz Hessiana es:

$$H = \begin{pmatrix} e^x(y^2+1) + \sin(x+y^2) & 2ye^x + 2y\sin(x+y^2) \\ 2ye^x + 2y\sin(x+y^2) & 2(e^x+1) - 2\cos(x+y^2) + 4y^2\sin(x+y^2) \end{pmatrix}$$

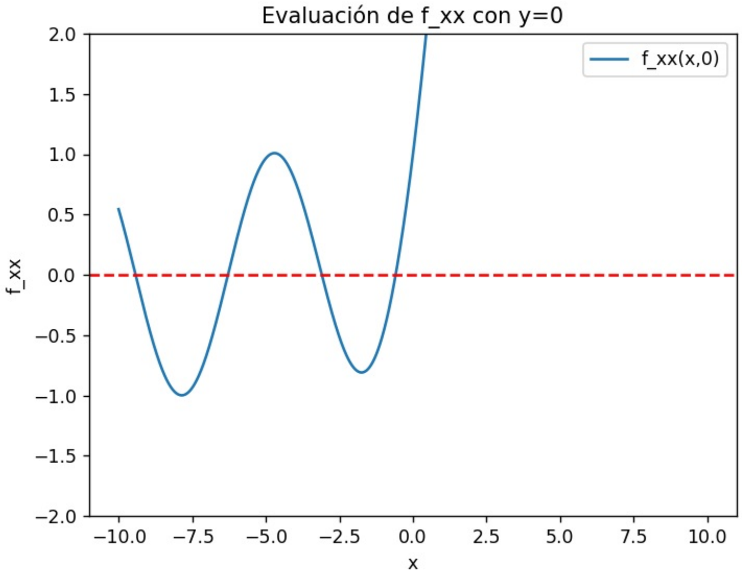
Condiciones para que la matriz Hessiana sea definida positiva:

Para una matriz 2×2 de la forma $H = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$, las condiciones son:

1. $a > 0$
2. $\det(H) = ac - b^2 > 0$

Análisis de la convexidad:

Analizaremos si la función $\frac{\partial^2 f}{\partial x^2} = e^x(y^2+1) + \sin(x+y^2)$ es positiva en todo su dominio. Consideremos los valores de dicha función cuando $y = 0$ y $x \in [-10, 10]$.



Como se puede observar en la gráfica, la función $\frac{\partial^2 f}{\partial x^2}$ en el punto $(-1, 0)$ toma un valor menor que cero:

$$\frac{\partial^2 f}{\partial x^2}(-1,0) = e^{-1}(0+1) + \sin(-1+0) = e^{-1} + \sin(-1) \approx 0{,}3679 - 0{,}8415 = -0{,}4736 < 0$$

Por tanto, la función **NO ES CONVEXA** en todo su dominio, ya que existe al menos un punto donde el elemento a de la matriz Hessiana es negativo.

Algoritmos de Optimización

1. Método de Región de Confianza

El método de Región de Confianza es un algoritmo de optimización no lineal que construye un modelo local de la función objetivo y restringe la búsqueda a una región donde el modelo es considerado confiable.

Según la conferencia, el método resuelve en cada iteración un subproblema de la forma:

$$\min_h \hat{f}(x+h) = f(x_k) + \nabla f(x_k)^T h + \frac{1}{2} h^T B_k h$$

sujeto a la restricción $\|h\| \leq \Delta_k$, donde Δ_k es el radio de la región de confianza y B_k es la matriz Hessiana o una aproximación de ella.

El algoritmo adapta dinámicamente el radio Δ_k basándose en la razón de reducción:

$$R_k = \frac{f(x_k) - f(x_{k+1})}{\hat{f}(x_k) - \hat{f}(x_{k+1})}$$

Si R_k es cercano a 1, el modelo es bueno y se puede aumentar el radio; si es cercano a 0 o negativo, el modelo es pobre y se debe reducir el radio.

2. Método de Regularización Adaptativa Cúbica (ARC)

El método de Regularización Adaptativa Cúbica (Adaptive Regularization with Cubics) es una extensión del enfoque de región de confianza que utiliza un modelo cúbico en lugar de cuadrático para aproximar la función objetivo.

El modelo cúbico tiene la forma:

$$m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p + \frac{\sigma_k}{3}\|p\|^3$$

donde $\sigma_k > 0$ es un parámetro de regularización que se ajusta adaptativamente y B_k es la matriz Hessiana o su aproximación.

A diferencia de la región de confianza que restringe explícitamente el paso mediante una norma, ARC penaliza los pasos grandes mediante el término cúbico $\frac{\sigma_k}{3}\|p\|^3$, lo que proporciona mayor flexibilidad para manejar regiones con curvatura negativa o planas.

El método ajusta σ_k basándose en la calidad del modelo, aumentándolo cuando el modelo es pobre y disminuyéndolo cuando el modelo es bueno, lo que garantiza convergencia global hacia puntos estacionarios incluso para funciones no convexas.

Justificación de la Selección de Métodos

La función objetivo de nuestro trabajo es:

$$f(x, y) = (e^x + 1)(y^2 + 1) - \sin(x + y^2) - x,$$

la cual es continua, no convexa y además existe la solución de la misma, con Hessiana disponible para evaluaciones de segundo orden.

Dadas estas características, seleccionamos los siguientes métodos:

1. **Región de Confianza**
2. **Regularización Adaptativa Cúbica (ARC)**

Estos algoritmos fueron seleccionados por los siguientes motivos:

- **Aprovechan la información de segundo orden**: Ambos métodos utilizan la matriz Hessiana o aproximaciones de ella (en nuestro caso usaremos la matriz Hessiana exacta), dadas las características de no linealidad y no convexidad de nuestra función el eso de la misma es crucial.
- **Manejan explícitamente la no convexidad**: A diferencia de otros métodos, tanto Región de Confianza como ARC incorporan mecanismos para controlar el paso cuando la Hessiana no es definida positiva, evitando convergencia a puntos de silla o direcciones de ascenso.
- **Control adaptativo del paso**: El ajuste dinámico del radio de confianza (en Región de Confianza) y del parámetro de regularización cúbica (en ARC) permite manejar eficientemente las oscilaciones introducidas por el término trigonométrico y las variaciones bruscas del término exponencial.
- **Especializados para problemas irrestrictos**: Están diseñados específicamente para optimización sin restricciones, que es el caso de nuestro problema.

Flujo de Trabajo, análisis de algoritmos y resultados

1. Flujo de Trabajo

El código desarrollado en `Algorithms.py` es el encargado de desarrollar todo lo necesario para analizar el comportamiento y los resultados de implementar y poner en práctica los algoritmos seleccionados. El código podemos decir que está dividido en 5 partes fundamentales que describiremos a continuación:

1. **Definición de la función objetivo y sus derivadas:**
   - `f(x)`: Calcula el valor de la función objetivo
   $$f(x, y) = (e^x + 1)(y^2 + 1) - \sin(x + y^2) - x$$
   - `grad_f(x)`: Evalúa el vector gradiente analítico
   $$\nabla f(x, y) = \begin{bmatrix} e^x(y^2 + 1) - \cos(x + y^2) - 1 \\ 2y(e^x + 1) - 2y\cos(x + y^2) \end{bmatrix}$$
   - `hess_f(x)`: Calcula la matriz Hessiana
   $$\nabla^2 f(x, y) = \begin{bmatrix} e^x(y^2 + 1) + \sin(x + y^2) & 2y(e^x + \sin(x + y^2)) \\ 2y(e^x + \sin(x + y^2)) & 2(e^x + 1 - \cos(x + y^2)) + 4y^2\sin(x + y^2) \end{bmatrix}$$

2. **Implementación del método de Región de Confianza:**
   - Inicializar con parámetros como punto inicial x_0, radio inicial Δ_0, umbral η y tolerancia.
   - En cada iteración realizamos lo siguiente:
     a) Calcular gradiente g y Hessiano B.
     b) Resolver el subproblema cuadrático para obtener el paso p (método de Newton o paso de Cauchy).
     c) Evaluar la razón de reducción
     $$\rho = \frac{f(x_k) - f(x_k + p)}{m_k(0) - m_k(p)}$$
     d) Ajustar el radio de la región de confianza Δ_k según ρ.
     e) Aceptar o rechazar el paso dependiendo del valor de η.
   - Almacenamos la trayectoria completa de puntos visitados.

3. **Implementación del método ARC (Adaptive Regularization with Cubics):**
   - Empleamos un modelo cúbico local:
   $$m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2}p^T B_k p + \frac{\sigma_k}{3}\|p\|^3$$
   - Resolver el sistema regularizado $(B + \sigma I)p = -g$.
   - Calcular la razón de reducción ρ y ajusta el parámetro de regularización σ_k:
     - Si $\rho < 0{,}25$, aumenta σ_k (pasos más conservadores).
     - Si $\rho > 0{,}75$, reduce σ_k (pasos más agresivos).
   - Actualizar el punto $x_{k+1} = x_k + p$ si el paso es aceptado.

4. **Ejecución y comparativa en múltiples rangos:**

- Se ejecutan ambos métodos con el mismo punto inicial x_0, generado aleatoriamente en tres intervalos distintos:

$$[-2, 2], \quad [-10, 10], \quad [-100, 100]$$

- Para cada rango, se almacena:
  - El historial de iteraciones.
  - Los valores de la función $f(x)$.
  - El punto óptimo final y la norma del gradiente.

5. **Visualización de resultados:**
   - Se construye una malla 2D para representar el contorno de la función objetivo.
   - Se grafican las trayectorias de ambos métodos sobre dicho contorno.
   - Se presenta un gráfico de convergencia en escala logarítmica que muestra la disminución de $f(x) - f^*$ a lo largo de las iteraciones.
   - Finalmente, se genera una tabla comparativa que resume los resultados para los tres rangos, incluyendo número de iteraciones, valores finales de $f(x)$ y norma del gradiente.

2. Resultados

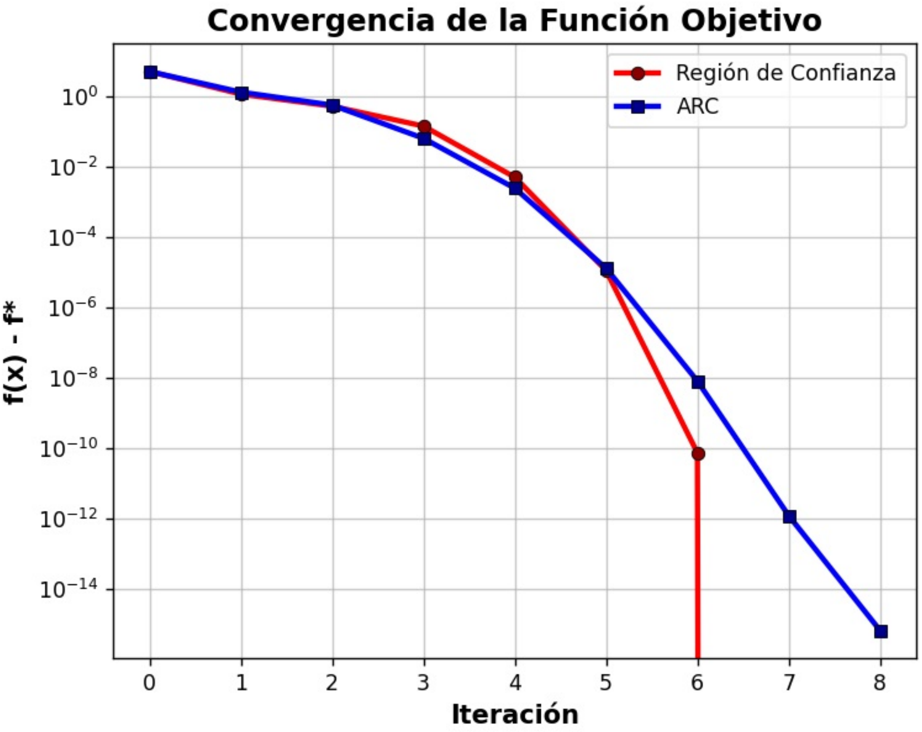
Rango	**Método**	**Iter.**	**f(x)**	$\|\nabla f\|$	**Punto óptimo**
[-2, 2]	TR	8	1.6575e+00	1.45e-10	(0.6013, 0.0000)
[-2, 2]	ARC	9	1.6575e+00	1.55e-08	(0.6013, 0.0000)
[-10, 10]	TR	1001	2.6404e+02	2.87e-01	(-4.8525, 16.0041)
[-10, 10]	ARC	1001	3.2099e+02	4.63e-01	(-5.0532, 17.6931)
[-100, 100]	TR	1000	8.1312e+03	3.80e-01	(-8.3089, 90.1108)
[-100, 100]	ARC	995	8.1061e+03	5.75e-02	(-8.3058, 89.9708)

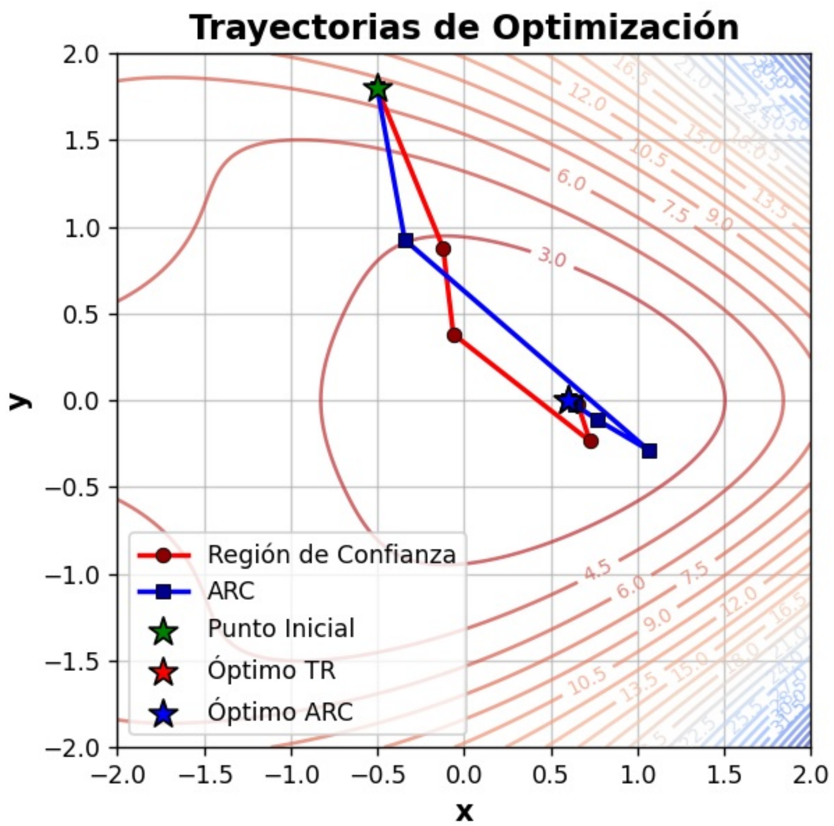
Analizando la Tabla anterior con los resultados obtenidos aplicando los algoritmos a los distintos rango podemos observar lo siguiente:

En el rango [-2, 2]:

- Ambos algoritmos demuestran un correcto desempeño en entornos controlados.
- El método de Región de Confianza (TR) converge en 8 iteraciones, mientras que ARC lo hace en 9, alcanzando ambos el mismo valor óptimo de la función $f(x) \approx 1{,}6575$.
- Las normas del gradiente en el punto final ($\approx 10^{-10}$ para TR y $\approx 10^{-8}$ para ARC) confirman que se alcanzó una región estacionaria de alta precisión.

- Esto indica que en problemas bien condicionados y cerca del óptimo, ambos métodos son equivalentes en precisión, con TR ligeramente más rápido en términos de iteraciones.



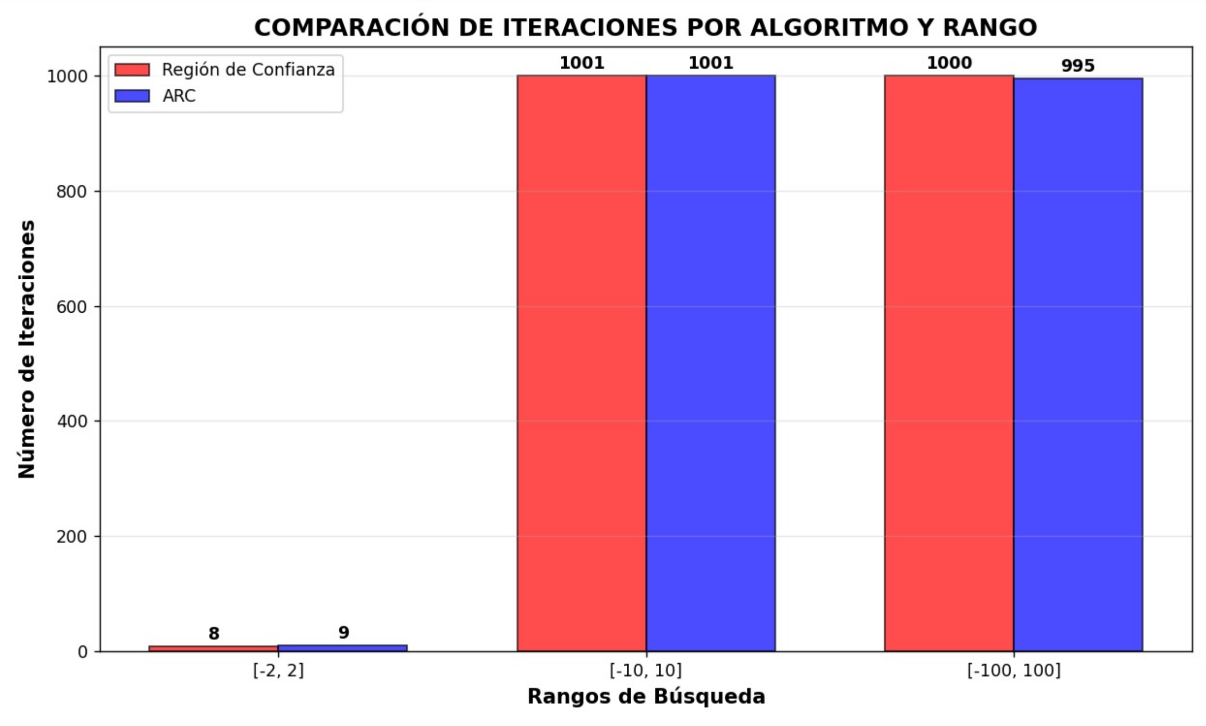


En el rango [-10, 10]:

- Ambos métodos encuentran dificultades significativas, no logrando converger dentro del límite de 1000 iteraciones.
- Los valores finales de $f(x)$ son considerablemente más altos (TR: 264.04, ARC: 320.99), sugiriendo que se quedaron atascados en mínimos locales o regiones planas. Esto ocurre porque la función objetivo presenta muchas curvas, valles y zonas planas, lo que provoca que los algoritmos se **estanquen** en puntos donde el gradiente es muy pequeño, aunque estos no correspondan al mínimo global. Esto se debe a que ambos métodos utilizan únicamente información local del punto actual (gradiente y Hessiano), por lo que carecen de la capacidad de escapar de dichos mínimos o mesetas.
- El método TR obtiene un valor de función ligeramente mejor y un gradiente final menor (0,287 vs 0,463 de ARC), este hecho podría indicar una mejor capacidad de exploración en este rango.
- Las coordenadas de los puntos óptimos encontrados difieren significativamente entre ambos métodos.

En el rango [-100, 100]:

- La complejidad del problema se acentúa notablemente.
- Ambos algoritmos exceden el límite de iteraciones, con valores de $f(x)$ muy elevados (TR: 8131.2, ARC: 8106.1).
- ARC muestra su ventaja en robustez numérica: aunque el valor de la función es solo ligeramente mejor que TR, la norma del gradiente final (0,0575) es significativamente menor que la de TR (0,380).
- Esto sugiere que ARC consigue acercarse más a una verdadera región estacionaria, gracias al efecto estabilizador del término cúbico en su formulación.



Tendencias generales:

- Se observa claramente que el desempeño de ambos algoritmos se degrada conforme ampliamos el rango de búsqueda inicial.
- Mientras que en entornos controlados son prácticamente equivalentes, en regiones más amplias ARC muestra mayor estabilidad numérica.
- Esta ventaja de ARC se evidencia particularmente por normas de gradiente más bajas en los rangos más extensos.
- Por otro lado, TR demuestra ser ligeramente más eficiente en el rango intermedio [-10,10], logrando un valor de función menor.

Análisis Comparativo de Estrategias de Optimización

Método de Región de Confianza (TR)	Método ARC (Regularización Cúbica Adaptativa)
Fortalezas Principales: • Precisión más exacta en entornos controlados • Convergencia más rápida en dominios acotados • Eficiencia computacional en problemas bien condicionados	**Fortalezas Principales:** • Robustez superior en paisajes complejos • Estabilidad numérica en regiones problemáticas • Mecanismo de protección contra pasos excesivos
Escenario Ideal: • Funciones suaves y predecibles • Espacios de solución moderados como el que analizamos anteriormente $[-2, 2]$ • Cuando la precisión es crítica • Problemas con curvaturas estables	**Escenario Ideal:** • Territorios de búsqueda extensos y hostiles • Dominios amplios y no acotado tales como $[-100, 100]$ • Cuando la robustez prevalece sobre precisión extrema • Funciones con comportamientos más irregulares
Característica Distintiva: Ajuste dinámico del radio de confianza	**Característica Distintiva:** Regularización cúbica adaptativa

Recomendaciones de Selección:

- **Región de Confianza (TR)** cuando se trabaje con problemas académicos bien definidos, funciones de prueba suaves, o cuando se requiera validación teórica con alta precisión numérica.
- **Regulación Adaptativa Cúbica (ARC)** en escenarios del mundo real con ruido numérico, problemas de gran escala, o cuando la función objetivo presenta comportamientos impredecibles y cambios bruscos de curvatura.