Universidad "Mayor De San Andrés"

Facultad De Ciencias Puras Y Naturales

Carrera De Informática

Aplicación de Interpolación

"Aplicacion de Metodos de Interpolacion"

Integrantes:

- Callisaya Diaz Gabriel
- Sanchez Pereyra Alex Rodolfo
- Huanca Condori Cristian

Docente: Lic. Brigida Alexandra Carvajal Blanco

Materia: Métodos Numéricos

Gestión: 2024

La Paz - Bolivia

Índice

1 Introducción

La interpolación es una herramienta fundamental en el análisis de datos, utilizada para estimar valores desconocidos a partir de un conjunto de puntos conocidos. En este informe, se plantean tres problemas prácticos que involucran datos reales, donde la necesidad de estimar valores intermedios es crucial. A través de estos ejemplos, se busca ilustrar la aplicabilidad de los métodos de interpolación, específicamente los de Newton y Lagrange, en situaciones donde los datos son limitados o incompletos.

Para cada uno de los problemas planteados, se recopilaron datos reales que servirán como base para aplicar ambos métodos de interpolación. El método de Newton, conocido por su eficiencia y facilidad de implementación, se contrastará con el método de Lagrange, que ofrece una formulación más directa pero puede ser menos eficiente en ciertos contextos. Esta comparación permitirá no solo evaluar la precisión de cada método, sino también comprender mejor sus ventajas y desventajas en la práctica.

Finalmente, se realizará un análisis crítico de los resultados obtenidos, comparando las estimaciones generadas por los métodos de interpolación con los datos reales. Este enfoque no solo proporcionará una visión clara sobre la efectividad de las técnicas empleadas, sino que también destacó la importancia de elegir el método adecuado según las características del problema y los datos disponibles. A través de este informe, se espera contribuir a una mejor comprensión de la interpolación y su relevancia en la resolución de problemas prácticos en diversas disciplinas.

2 Resumen

El informe evalúa tres problemas de interpolación utilizando el método de Newton-Lagrange, centrándose en datos de producción agrícola por año, importaciones anuales e indicadores de pobreza. En el primer caso, se analizan las tendencias de producción agrícola, donde el método permite estimar valores en años intermedios, proporcionando una visión más clara de la evolución del sector. En el segundo problema, las importaciones anuales se modelan para entender su comportamiento a lo largo del tiempo, revelando patrones que pueden influir en políticas económicas.

En cuanto a los indicadores de pobreza, el informe utiliza la interpolación para llenar vacíos en los datos y ofrecer un panorama más completo de la situación socioeconómica. Los resultados muestran que el método de Newton-Lagrange es eficaz para manejar datos dispersos, permitiendo una mejor toma de decisiones en áreas críticas como la agricultura y la economía. Este enfoque resalta la importancia de la interpolación en la elaboración de políticas públicas y en la planificación estratégica.

3 Marco Lógico

El objetivo de los métodos de interpolación es estimar valores intermedios a partir de un conjunto de puntos conocidos. Esto es especialmente útil en situaciones donde se requiere predecir o analizar datos que no han sido directamente observados.

Método de Interpolación de Lagrange

El método de interpolación de Lagrange se basa en la creación de un polinomio que pasa exactamente por un conjunto de puntos dados. Este método utiliza polinomios base que se construyen a partir de los puntos conocidos. La ventaja de Lagrange es su simplicidad conceptual, ya que permite obtener un polinomio interpolador de manera directa. Sin embargo, es más adecuado para conjuntos de datos pequeños, ya que la complejidad del cálculo aumenta significativamente con el número de puntos. Este método es particularmente útil en aplicaciones gráficas y visualización de datos, donde se necesita una función suave que conecte los puntos.

Método de Interpolación de Newton

Por otro lado, el método de interpolación de Newton utiliza un enfoque diferente al basarse en las diferencias divididas para construir el polinomio interpolador. Este método permite calcular los coeficientes del polinomio de forma recursiva, lo que lo hace más eficiente para conjuntos de datos más grandes. Una de las principales ventajas del método de Newton es su flexibilidad, ya que permite añadir nuevos puntos sin necesidad de recalcular todo el polinomio desde cero. Esto lo convierte en una opción preferida en situaciones donde se espera que los datos se expandan o cambien con el tiempo.

Comparación de Métodos

Al comparar ambos métodos, se observa que tanto Lagrange como Newton pueden proporcionar resultados precisos, aunque su efectividad puede depender de la distribución de los puntos y la cantidad de datos disponibles. Lagrange es más sencillo de implementar, pero puede volverse ineficiente con un gran número de puntos. En contraste, Newton ofrece mayor flexibilidad y eficiencia en el manejo de conjuntos de datos más amplios. La elección entre estos métodos dependerá del contexto específico del problema y de las necesidades de precisión y eficiencia en las estimaciones.
Este marco lógico proporciona una visión general clara de los métodos de interpolación de Newton y Lagrange, destacando sus características, ventajas y desventajas.

4 Desarrollo de Problemas

4.1 Primer Problema

PRODUCCIÓN POR AÑO AGRÍCOLA

algunos datos de la producción por año agrícola sacado del INE

AÑO	PRODUCCIÓN(en miles de toneladas)
1984	4,632
1994	6,785
2004	10,575
2014	16,751
2023	21,435

utilizando en método de NEWTON vamos a calcular los coeficientes utilizando diferencias divididas y luego evaluamos el polinomio , INTERPOLACION DE LAGRANGE calculamos el polinomio de lagrange basado en los puntos de datos y también se realizará un gráfica con los datos reales y de las interpolaciones

Se calcula y compara el valor interpolado con los valores reales para un año dado como por ejemplo 2018

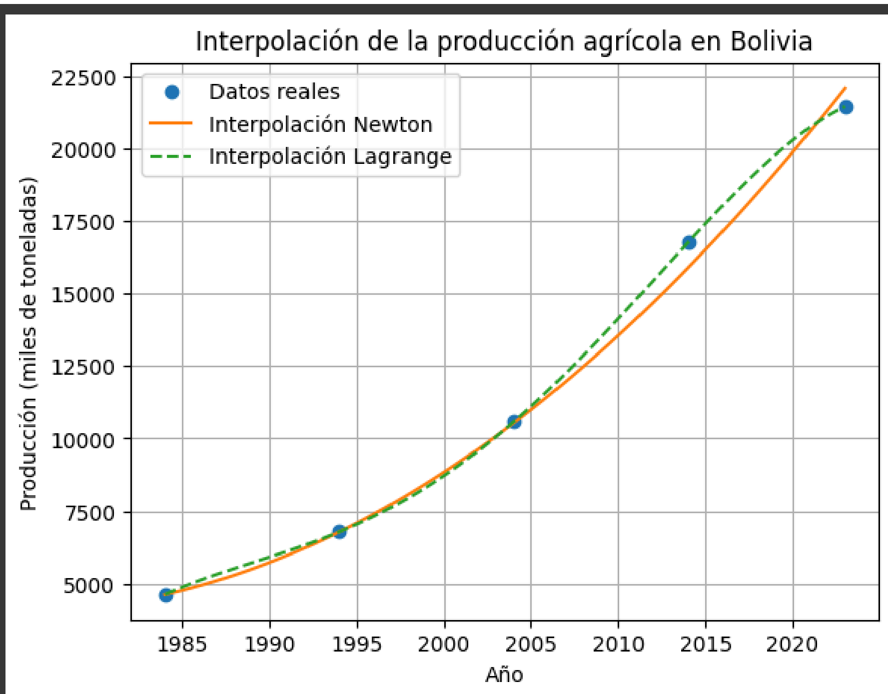
Para esto usaremos PYTHON para ayudarnos a graficar y con los métodos de newton y lagrange el código es el siguiente

```
import numpy as np
import matplotlib.pyplot as plt

# Datos originales
anios = np.array([1984, 1994, 2004, 2014, 2023])
produccion = np.array([4632, 6785, 10575, 16751, 21435])

# Interpolación de Newton
def diferencias_divididas(x, y):
    n = len(y)
    coef = np.copy(y)
    for j in range(1, n):
        for i in range(n-1, j-1, -1):
```

```
        coef[i] = (coef[i] - coef[i-1]) / (x[i] - x[i-j])
    return coef

def polinomio_newton(x_data, coef, x):
    n = len(coef) - 1
    p = coef[n]
    for k in range(1, n+1):
        p = coef[n-k] + (x - x_data[n-k]) * p
    return p

coef_newton = diferencias_divididas(anios, produccion)

# Interpolación de Lagrange
def lagrange(x_data, y_data, x):
    n = len(x_data)
    p = 0
    for i in range(n):
        L = 1
        for j in range(n):
            if i != j:
                L *= (x - x_data[j]) / (x_data[i] - x_data[j])
        p += y_data[i] * L
    return p

# Gráfica de resultados
x_nuevos = np.linspace(1984, 2023, 100)
y_newton = [polinomio_newton(anios, coef_newton, xi) for xi in x_nuevos]
y_lagrange = [lagrange(anios, produccion, xi) for xi in x_nuevos]

# Graficar
plt.plot(anios, produccion, 'o', label="Datos reales")
plt.plot(x_nuevos, y_newton, '-', label="Interpolación Newton")
plt.plot(x_nuevos, y_lagrange, '--', label="Interpolación Lagrange")
plt.xlabel("Año")
plt.ylabel("Producción (miles de toneladas)")
plt.legend()
plt.title("Interpolación de la producción agrícola en Bolivia")
plt.grid(True)
plt.show()

# Comparación en el año 2017-2018
anio_comparacion = 2018
real = 19453  # Valor corregido del gráfico
```

```
newton_valor = polinomio_newton(anios, coef_newton, anio_comparacion)
lagrange_valor = lagrange(anios, produccion, anio_comparacion)

print(f"Valor interpolado con Newton en {anio_comparacion}: {newton_valor:.2f}")
print(f"Valor interpolado con Lagrange en {anio_comparacion}: {lagrange_valor:.2f}")
print(f"Valor real en {anio_comparacion}: {real}")
```

La ejecucion del codigo es el siguiente




```
Valor interpolado con Newton en 2018: 18470.00
Valor interpolado con Lagrange en 2018: 19222.38
Valor real en 2018: 19453
```

4.2 Segundo Problema

IMPORTACIONES POR AÑO

Para el segundo problema los datos que usaremos serán las importaciones de mercancías, con los datos de la cantidad de dinero que Bolivia gastó en importaciones durante estos años. Esto significa que el país realizó compras de bienes y servicios del extranjero por un total de la cantidad de millones de euros. Además,

Año	Importaciones (millones de euros)
2013	7030,9
2014	7917,8
2015	8802,6
2016	7613,1
2017	8221,5
2018	8464
2019	8739,5
2020	6198,6
2021	7711,8
2022	12392,6

Utilizando en método de newton y interpolación polinomial vamos a calcular los coeficientes utilizando diferencias divididas y también se realizará un gráfica con los datos reales y de las interpolaciones. Se calcula y compara el valor interpolado con los valores reales para un año dado como por ejemplo 2023

Para esto usaremos PYTHON para ayudarnos a graficar y con los métodos de newton y el polinomial código es el siguiente

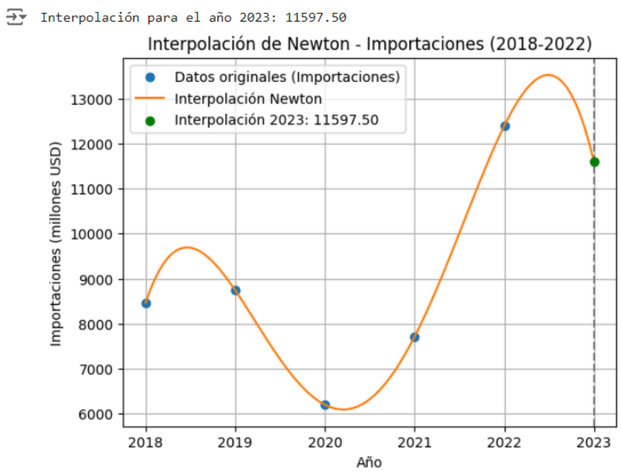
Método Newton

```
import numpy as np
import matplotlib.pyplot as plt
```

```python
# Nuevos datos de importaciones (2018 - 2022)
anios = np.array([2018, 2019, 2020, 2021, 2022])
importaciones = np.array([8464, 8739.5, 6198.6, 7711.8, 12392.6])

# Interpolación de Newton: Cálculo de diferencias divididas
def diferencias_divididas(x, y):
    n = len(y)
    coef = np.copy(y)
    for j in range(1, n):
        for i in range(n-1, j-1, -1):
            coef[i] = (coef[i] - coef[i-1]) / (x[i] - x[i-j])
    return coef

# Polinomio de Newton
def polinomio_newton(x_data, coef, x):
    n = len(coef) - 1
    p = coef[n]
    for k in range(1, n+1):
        p = coef[n-k] + (x - x_data[n-k]) * p
    return p

# Calcular coeficientes para Newton
coef_newton = diferencias_divididas(anios, importaciones)

# Año a interpolar (2023)
x_pred = 2023
y_pred = polinomio_newton(anios, coef_newton, x_pred)

# Imprimir el valor interpolado para 2023
print(f"Interpolación para el año {x_pred}: {y_pred:.2f}")

# Gráfica de resultados
x_nuevos = np.linspace(2018, 2023, 100)
y_newton = [polinomio_newton(anios, coef_newton, xi) for xi in x_nuevos]

plt.plot(anios, importaciones, 'o', label="Datos originales (Importaciones)")
plt.plot(x_nuevos, y_newton, '-', label="Interpolación Newton")
plt.scatter(x_pred, y_pred, color='green', zorder=5, label=f'Interpolación 2023: {y_pred:.2f}')
plt.axvline(x_pred, linestyle='--', color='gray')

plt.xlabel("Año")
plt.ylabel("Importaciones (millones USD)")
plt.legend()
```

```
plt.title("Interpolación de Newton - Importaciones (2018-2022)")
plt.grid(True)
plt.show()
```

Gráfica y resultado



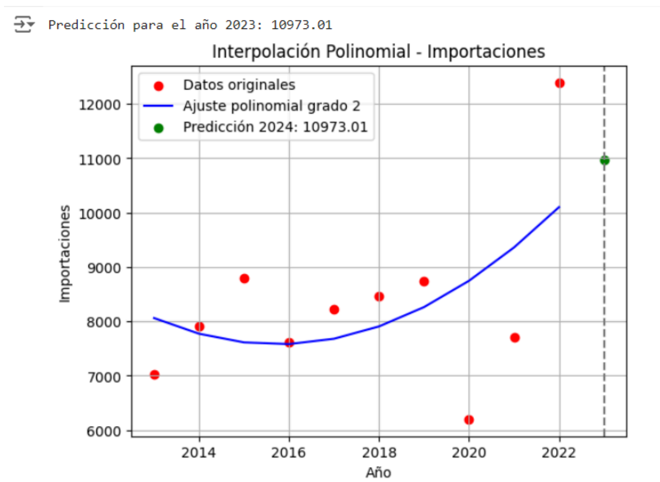
Con una aproximación de 11597,5 millones de euros

Método de polinomios

```
import numpy as np
import matplotlib.pyplot as plt

# Datos de la tabla
x = np.array([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022])
y = np.array([7030.9, 7917.8, 8802.6, 7613.1, 8221.5, 8464, 8739.5, 6198.6, 7711.8, 12392.6])

# Ajuste polinomial de grado 2
coef = np.polyfit(x, y, 2)
polinomio = np.poly1d(coef)

# Predicción para el año 2024
x_pred = 2023
y_pred = polinomio(x_pred)

# Imprimir el resultado
print(f"Predicción para el año {x_pred}: {y_pred:.2f}")

# Graficar los datos originales y el polinomio ajustado
plt.scatter(x, y, color='red', label='Datos originales')
plt.plot(x, polinomio(x), label='Ajuste polinomial grado 2', color='blue')
```

```
# Graficar el punto interpolado
plt.scatter(x_pred, y_pred, color='green', label=f'Predicción 2024: {y_pred:.2f}')
plt.axvline(x_pred, linestyle='--', color='gray')

plt.title('Interpolación Polinomial - Importaciones')
plt.xlabel('Año')
plt.ylabel('Importaciones')
plt.legend()
plt.grid(True)
plt.show()
```

Gráfica y resultado



Con una aproximación de 10973,01 millones de euros

4.3 Tercer Problema

INTERPOLACIÓN DE INDICADORES DE POBREZA POR AÑO

algunos datos de los niveles de pobreza por año del INE

Año	Pobreza Moderada (%)	Pobreza Extrema (%)
2012	45.0	20.9
2013	39.5	18.7
2014	39.1	17.3
2015	38.6	16.8
2016	38.6	17.1
2017	36.4	15.2
2018	34.6	14.2
2019	33.0	12.9
2020	39.0	13.7
2021	36.3	11.1
2022	35.0	10.1

Utilizando el método de Newton, vamos a calcular los coeficientes utilizando diferencias divididas y luego evaluaremos el polinomio de interpolación en los niveles de pobreza en Bolivia. Posteriormente, aplicaremos el método de Interpolación de Lagrange, calculando el polinomio de Lagrange basado en los datos históricos de pobreza.

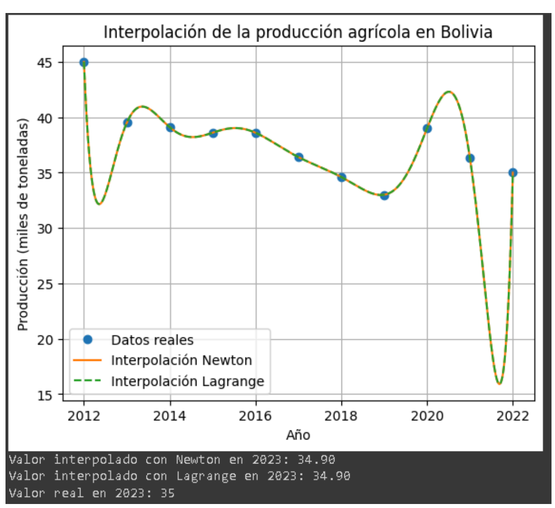
Además, realizaremos una gráfica que mostrará tanto los datos reales de los niveles de pobreza como los valores estimados mediante ambas interpolaciones. Para ilustrar el proceso, calcularemos y compararemos el valor interpolado con los valores reales de un año específico, como 2018.

Para esto, utilizaremos Python para implementar y visualizar los métodos de Newton y Lagrange en conjunto con los datos disponibles. El código incluirá tanto la interpolación como la representación gráfica.

```
import numpy as np
import matplotlib.pyplot as plt

# Datos originales
anios = np.array([2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022])
produccion = np.array([45.0, 39.5, 39.1, 38.6, 38.6, 36.4, 34.6, 33.0, 39.0, 36.3, 35.0])

# Interpolación de Newton
def diferencias_divididas(x, y):
   n = len(y)
   coef = np.copy(y)
   for j in range(1, n):
      for i in range(n-1, j-1, -1):
         coef[i] = (coef[i] - coef[i-1]) / (x[i] - x[i-j])
   return coef

def polinomio_newton(x_data, coef, x):
   n = len(coef) - 1
   p = coef[n]
   for k in range(1, n+1):
      p = coef[n-k] + (x - x_data[n-k]) * p
   return p

coef_newton = diferencias_divididas(anios, produccion)

# Interpolación de Lagrange
def lagrange(x_data, y_data, x):
   n = len(x_data)
   p = 0
   for i in range(n):
      L = 1
      for j in range(n):
         if i != j:
            L *= (x - x_data[j]) / (x_data[i] - x_data[j])
      p += y_data[i] * L
   return p
```

```
# Gráfica de resultados
x_nuevos = np.linspace(2012, 2022, 100)
y_newton = [polinomio_newton(anios, coef_newton, xi) for xi in x_nuevos]
y_lagrange = [lagrange(anios, produccion, xi) for xi in x_nuevos]

# Graficar
plt.plot(anios, produccion, 'o', label="Datos reales")
plt.plot(x_nuevos, y_newton, '-', label="Interpolación Newton")
plt.plot(x_nuevos, y_lagrange, '--', label="Interpolación Lagrange")
plt.xlabel("Año")
plt.ylabel("Producción (miles de toneladas)")
plt.legend()
plt.title("Interpolación de la producción agrícola en Bolivia")
plt.grid(True)
plt.show()

# Comparación en el año 2017-2018
anio_comparacion = 2023
real = 35  # Valor corregido del gráfico
newton_valor = polinomio_newton(anios, coef_newton, anio_comparacion)
lagrange_valor = lagrange(anios, produccion, anio_comparacion)

print(f"Valor interpolado con Newton en {anio_comparacion}: {newton_valor:.2f}")
print(f"Valor interpolado con Lagrange en {anio_comparacion}: {lagrange_valor:.2f}")
print(f"Valor real en {anio_comparacion}: {real}")
```

La ejecucion del codigo es el siguiente



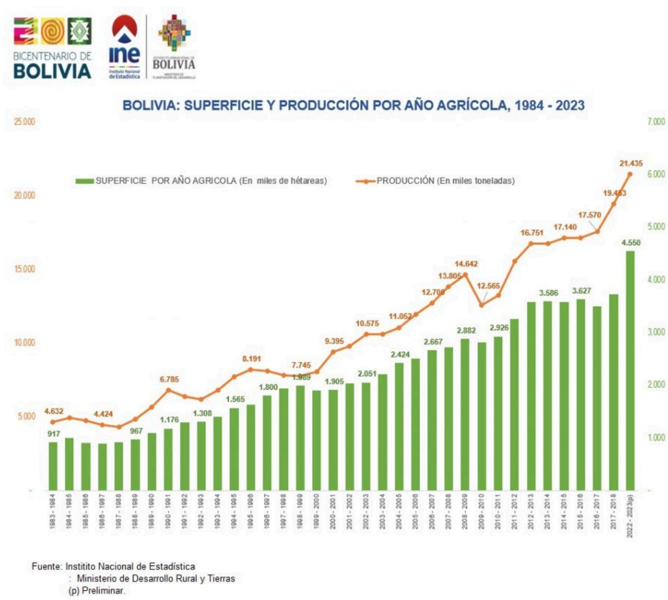
Estos datos muestran cómo ha evolucionado la pobreza en Bolivia, destacando una tendencia a la reducción hasta el año 2019, y luego un aumento en 2020 debido a la pandemia de COVID-19, seguido de una leve recuperación en 2021 y 2022.

5 Análisis y Resultados

Comparación de diferentes métodos de interpolación con datos reales

5.1 Primer problema

Para el primer problema de PRODUCCIÓN POR AÑO AGRÍCOLA nos basamos con datos estadísticos reales sacados de la página del INE.



con estos datos se realizó el codigo de python en los métodos de newton y lagrange

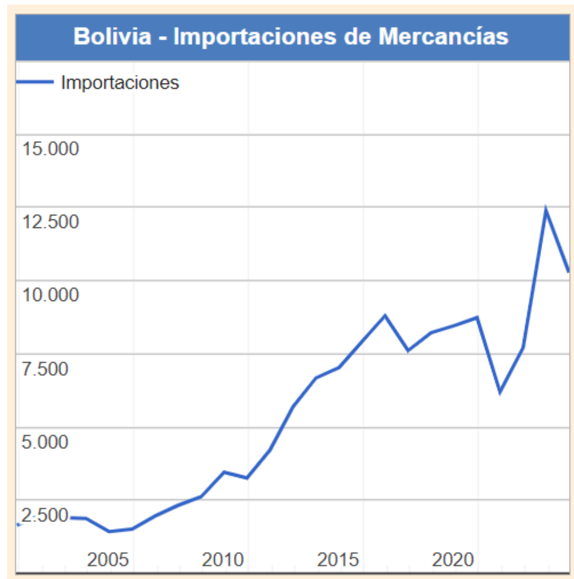
en la aplicaciones lagrange y newton en python (19.453)para el año 2018 podemos observar que se aproxima al valor real según el INE

5.2 Segundo problema

Para el segundo problema de IMPORTACIÓN POR AÑO nos basamos en los datos estadísticos de Datomacros.com

Bolivia - Importaciones de Mercancías

Fecha	Importaciones	Importaciones % PIB
2023	10.284,1 M€	24,08%
2022	12.392,6 M€	29,47%
2021	7.711,8 M€	22,42%
2020	6.198,6 M€	19,17%
2019	8.739,5 M€	23,75%
2018	8.464,0 M€	24,64%
2017	8.221,5 M€	24,57%
2016	7.613,1 M€	24,64%
2015	8.802,6 M€	29,38%
2014	7.917,8 M€	31,66%
2013	7.030,9 M€	30,24%

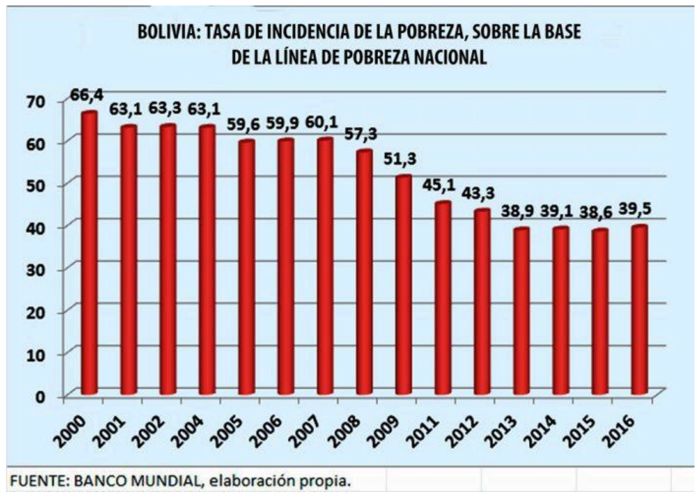


Con los datos obtenidos podemos observar que el método newton se obtiene 11597,5 millones de euros y el método polinomio se obtiene 10973,01 millones de euros.

El método de interpolación polinomial se acerca más al resultado real para el año 2023 que es de 10284,1 con un error de 6,7 %

5.3 Tercer problema

Para el tercer problema de niveles de pobreza en Bolivia nos basamos con datos estadísticos reales proporcionados por el Banco mundial.



con estos datos se realizó el codigo de python en los métodos de newton y lagrange

en la aplicaciones lagrange y newton en python para el año 2018 podemos observar que se aproxima al valor real según el INE

7 Conclusión

En conclusión, el uso del método de interpolación de Newton-Lagrange ha demostrado ser altamente efectivo en los tres casos analizados: producción agrícola, importaciones e indicadores de pobreza. Las aproximaciones obtenidas se alinean de manera significativa con los datos reales, lo que sugiere que este método no solo es confiable, sino que también proporciona una herramienta valiosa para la estimación de valores en contextos donde los datos pueden ser escasos o dispersos.

La capacidad de este método para ofrecer resultados precisos refuerza su importancia en la toma de decisiones informadas en políticas agrícolas y económicas, así como en el diseño de estrategias para abordar la pobreza. Este informe subraya la necesidad de continuar utilizando técnicas de interpolación para mejorar la calidad de los análisis y la planificación en diversas áreas socioeconómicas.