Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente:

Ing. Torres Cruz Fred

Alumno:

Ticona Miramira Roberto Angel

Trabajo 6 - Método de Punto Fijo

» Descripción

El método de Punto Fijo es una técnica iterativa utilizada para encontrar soluciones aproximadas a ecuaciones no lineales de la forma $f(x) = 0$. Su principio se basa en reescribir la ecuación en una forma equivalente:

$$x = g(x)$$

De esta manera, la raíz buscada es un punto donde la función $g(x)$ y la recta $y = x$ se intersectan, es decir, el punto donde $x = g(x)$. A partir de un valor inicial x_0, el método genera una sucesión de aproximaciones mediante la ecuación iterativa:

$$x_{n+1} = g(x_n)$$

El proceso continúa hasta que la diferencia entre dos iteraciones sucesivas sea menor que una tolerancia predefinida, lo que indica que se ha alcanzado una convergencia aceptable.

Condiciones de convergencia

Para que el método converja hacia una raíz real, se deben cumplir ciertas condiciones:

- La función $g(x)$ debe ser continua y diferenciable en un intervalo que contenga la raíz.
- Debe cumplirse que $|g'(x)| < 1$ en dicho intervalo para asegurar la convergencia.

Si estas condiciones no se cumplen, el método puede divergir o entrar en oscilaciones sin acercarse a la raíz.

Importancia de la visualización previa

Antes de aplicar el método, se recomienda graficar tanto la función original $f(x)$ como su forma iterativa $g(x)$ en un intervalo adecuado. Esta representación permite identificar las zonas donde $g(x)$ cruza la recta $y = x$, lo que sugiere posibles puntos fijos y ayuda a seleccionar un valor inicial x_0 apropiado para garantizar la convergencia.

Restricciones y limitaciones

- No siempre es posible encontrar una función $g(x)$ adecuada para una ecuación dada.
- La elección inadecuada de $g(x)$ o del valor inicial x_0 puede provocar divergencia.
- Puede requerir un número considerable de iteraciones si $|g'(x)|$ es cercano a 1.

» Aplicación del método en Python

El objetivo fue desarrollar un programa en Python que permita ingresar una función $f(x)$ y su forma iterativa $g(x)$, graficar la función y luego aplicar el método de Punto Fijo si el usuario lo desea. El programa calcula sucesivamente los valores de $x_{n+1} = g(x_n)$ hasta alcanzar una precisión deseada y muestra los resultados obtenidos.

» Entrada

- Una cadena de texto que representa la función original $f(x)$.
- La forma iterativa correspondiente $g(x)$.
- El intervalo para graficar.
- El valor inicial x_0.

$$f(x) = \cos(x) - x$$
$$g(x) = \cos(x)$$

» Salida

- Gráfica de la función $f(x)$.
- La raíz aproximada encontrada.
- Número de iteraciones realizadas hasta cumplir la tolerancia.

» Restricciones

- El método requiere que $|g'(x)| < 1$ en el entorno de la raíz para garantizar la convergencia.
- La función debe ser continua y evaluable en todo el intervalo seleccionado.

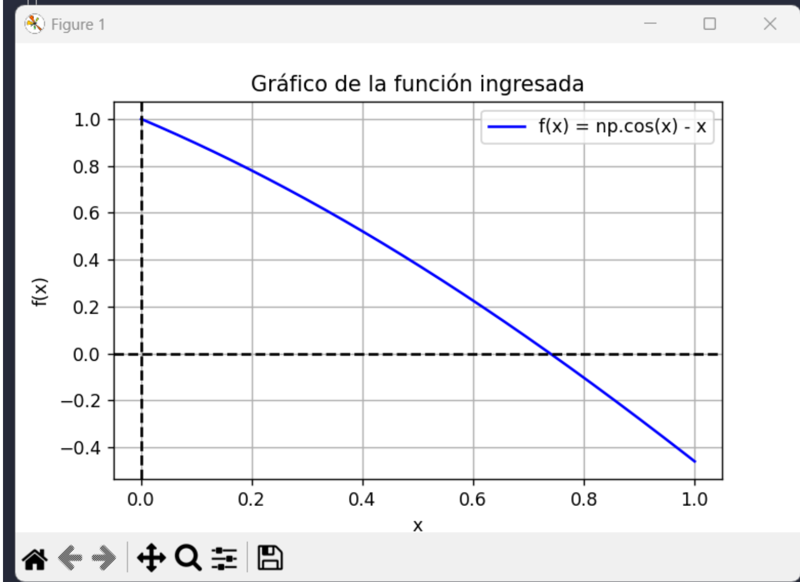
Código

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

# === 1. Ingreso de la función y su forma iterativa g(x) ===
print("=== MÉTODO DEL PUNTO FIJO ===")
print("Recuerde que f(x) = 0 se reescribe como x = g(x)")
print("Ejemplo: Si f(x) = cos(x) - x, entonces g(x) = cos(x)\n")

func_str = input("Ingrese la función original f(x): ")   # Ejemplo: np.cos(x) - x
g_str = input("Ingrese la función iterativa g(x): ")     # Ejemplo: np.cos(x)

# Definición de funciones
def f(x):
    return eval(func_str, {"np": np, "x": x})

def g(x):
    return eval(g_str, {"np": np, "x": x})

# === 2. Graficar la función f(x) ===
xmin = float(input("Ingrese el valor mínimo de x: "))
xmax = float(input("Ingrese el valor máximo de x: "))

x = np.linspace(xmin, xmax, 400)
y = f(x)

```

```python
plt.figure(figsize=(8, 5))
plt.plot(x, y, label=f"f(x) = {func_str}", color='blue')
plt.axhline(0, color='black', linestyle='--')
plt.axvline(0, color='black', linestyle='--')
plt.title("Gráfico de la función ingresada")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend()
plt.grid(True)
plt.show()

# === 3. Pregunta si desea aplicar el método de punto fijo ===
op = input("¿Desea aplicar el método de Punto Fijo? (s/n): ").lower()

if op == "s":
    # === 4. Ingreso de valor inicial ===
    x0 = float(input("Ingrese el valor inicial x0: "))

    tol = 1e-6
    max_iter = 100

    print("\nIteración |      x0       |      g(x0)      |      f(x0)      |   Error")
    print("---------------------------------------------------------------")

    for i in range(1, max_iter + 1):
        x1 = g(x0)
        error = abs(x1 - x0)

        print(f"{i:9d} | {x0:10.6f} | {x1:12.6f} | {f(x1):12.6f} | {error:10.6f}")

        if error < tol:
            print(f"\n Raíz aproximada encontrada: {x1:.6f}")
            print(f"Iteraciones realizadas: {i}")
            break

        x0 = x1
    else:
        print("\n No se alcanzó la convergencia después de", max_iter, "iteraciones.")
else:
    print("No se aplicó el método de Punto Fijo.")
```

Ejecución

```
=== MÉTODO DEL PUNTO FIJO ===
Recuerde que f(x) = 0 se reescribe como x = g(x)
Ejemplo: Si f(x) = cos(x) - x, entonces g(x) = cos(x)

Ingrese la función original f(x): np.cos(x) - x
Ingrese la función iterativa g(x): np:cos(x)
Ingrese el valor mínimo de x: 0
Ingrese el valor máximo de x: 1
```




```
=== MÉTODO DEL PUNTO FIJO ===
Recuerde que f(x) = 0 se reescribe como x = g(x)
Ejemplo: Si f(x) = cos(x) - x, entonces g(x) = cos(x)

Ingrese la función original f(x): np.cos(x) - x
Ingrese la función iterativa g(x): np.cos(x)
Ingrese el valor mínimo de x: 0
Ingrese el valor máximo de x: 1
¿Desea aplicar el método de Punto Fijo? (s/n): s
Ingrese el valor inicial x0: 0.5

Iteración |    x0     |   g(x0)    |   f(x0)    |  Error
-------------------------------------------------------------
        1 |  0.500000 |   0.877583 |  -0.238570 |  0.377583
        2 |  0.877583 |   0.639012 |   0.163673 |  0.238570
        3 |  0.639012 |   0.802685 |  -0.107907 |  0.163673
        4 |  0.802685 |   0.694778 |   0.073418 |  0.107907
        5 |  0.694778 |   0.768196 |  -0.049030 |  0.073418
        6 |  0.768196 |   0.719165 |   0.033190 |  0.049030
        7 |  0.719165 |   0.752356 |  -0.022275 |  0.033190
        8 |  0.752356 |   0.730081 |   0.015039 |  0.022275
        9 |  0.730081 |   0.745120 |  -0.010114 |  0.015039
       10 |  0.745120 |   0.735006 |   0.006820 |  0.010114
       11 |  0.735006 |   0.741827 |  -0.004591 |  0.006820
       12 |  0.741827 |   0.737236 |   0.003094 |  0.004591
       13 |  0.737236 |   0.740330 |  -0.002083 |  0.003094
       14 |  0.740330 |   0.738246 |   0.001404 |  0.002083
       15 |  0.738246 |   0.739650 |  -0.000945 |  0.001404
       16 |  0.739650 |   0.738705 |   0.000637 |  0.000945
       17 |  0.738705 |   0.739341 |  -0.000429 |  0.000637
       18 |  0.739341 |   0.738912 |   0.000289 |  0.000429
       19 |  0.738912 |   0.739201 |  -0.000195 |  0.000289
       20 |  0.739201 |   0.739007 |   0.000131 |  0.000195
```

```
    21 |  0.739007 |  0.739138 |  -0.000088 |  0.000131
    22 |  0.739138 |  0.739050 |   0.000060 |  0.000088
    23 |  0.739050 |  0.739109 |  -0.000040 |  0.000060
    24 |  0.739109 |  0.739069 |   0.000027 |  0.000040
    25 |  0.739069 |  0.739096 |  -0.000018 |  0.000027
    26 |  0.739096 |  0.739078 |   0.000012 |  0.000018
    27 |  0.739078 |  0.739090 |  -0.000008 |  0.000012
    28 |  0.739090 |  0.739082 |   0.000006 |  0.000008
    29 |  0.739082 |  0.739087 |  -0.000004 |  0.000006
    30 |  0.739087 |  0.739084 |   0.000003 |  0.000004
    31 |  0.739084 |  0.739086 |  -0.000002 |  0.000003
    32 |  0.739086 |  0.739084 |   0.000001 |  0.000002
    33 |  0.739084 |  0.739086 |  -0.000001 |  0.000001
    34 |  0.739086 |  0.739085 |   0.000001 |  0.000001

✅ Raíz aproximada encontrada: 0.739085
Iteraciones realizadas: 34
```