REGULAR METHODS FOR APPROXIMATE SOLUTIONS OF FIRST-TYPE INTEGRAL EQUATIONS

Namangan State University

f.m.f.n., associate professor Imomov A.

Namangan State University, Master's student

Nuriddinova (Pulatova) Mushtariy Shuhratjon qizi

Annotation: Integral equations are an important part of mathematical analysis and are widely used in various fields, including physics, economics, and other scientific disciplines. Integral equations are similar to differential equations, but their solutions are not directly determined. Instead, the solution depends on the integral of an unknown function. These equations are often used to model various physical processes such as heat conduction, electromagnetic waves, and acoustic phenomena.

Key words: Integral equations, Laplace Transform, Neumann Method, Numerical Methods, python, Simpson's Rule.

Integral equations can be divided into two main types: **first-order** and **second-order** integral equations. First-order integral equations have relatively simple forms and can be solved using either analytical or numerical methods. In contrast, second-order integral equations are more complex and require more sophisticated techniques for solving. This article will focus on **first-order integral equations** and their solution methods, providing examples and detailed explanations.

Methods

First-order integral equations are generally expressed in the following form:

$$f(x)=\lambda\int_a^b K(x,t)g(t)dt+h(x)$$

Here:

- $f(x)$ and $h(x)$ are given functions,
- $K(x,t)$ is the kernel function,
- $g(t)$ is the unknown function,

- λ is a parameter.

Several methods can be used to solve first-order integral equations:

1. Laplace Transform

The Laplace transform is a powerful method for transforming integral equations into differential equations, which are often easier to solve. When the kernel function $K(x,t)$ is simple, the Laplace transform can be used to obtain an analytical solution.

2. Neumann Method

The Neumann method is based on iterating the solution of the integral equation. Each iteration improves upon the previous approximation, providing a more accurate solution. This method is particularly useful when the kernel function is complex or when an exact analytical solution is difficult to obtain.

3. Numerical Methods

When analytical methods are not feasible, numerical integration methods such as **Simpson's Rule** or the **Trapezoidal Rule** can be used to compute approximate solutions to integral equations. These methods provide an efficient way to solve integral equations when an exact solution is hard to derive.

Example and Solution in Python

Let’s now consider a **first-order integral equation** and solve it using Python.

The equation is given by:

$$f(x)=\int_0^x (x-t)e^{-t}dt+x$$

Here, e^{-t}— is the function $g(t)$, and $(x-t)$ is the kernel function. We need to compute $f(x)$.

To solve this equation numerically, we will use **Simpson's Rule**, which is a higher-accuracy numerical integration method.

python

import numpy as np

```
import scipy.integrate as integrate
import matplotlib.pyplot as plt
                    # Kernel function and g(t) function
def kernel(x, t):
    return x - t
def g(t):
    return np.exp(-t)
                    # Define the integral function
def integral_function(x):
    result, _ = integrate.quad(lambda t: kernel(x, t) * g(t), 0, x)
    return result + x
                    # Generate values of f(x) for plotting
x_values = np.linspace(0.1, 10, 100)
f_values = [integral_function(x) for x in x_values]
                    # Plot the results
plt.plot(x_values, f_values, label="f(x) = ∫(x - t)e^(-t) dt + x")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.title("Solution of 1st Order Integral Equation")
plt.legend()
plt.grid(True)
plt.show()
```

Explanation and Results

The Python program computes the solution to the first-order integral equation and plots the results. **Simpson's Rule** is used to compute the integral numerically, and the **scipy.integrate.quad** function is employed to ensure high accuracy in the integral computation.

The plot displays how $f(x)$ varies with xxx, providing a visual representation of the solution to the integral equation. Numerical methods, such as the one used here, are particularly valuable when the kernel or $g(t)$ is complex and an exact analytical solution is difficult to obtain.

Conclusion

First-order integral equations can be solved using various mathematical methods, depending on the complexity of the equation. Analytical methods, such as Laplace transforms or the Neumann method, work well for simpler cases. However, for more complex equations, numerical methods, such as **Simpson's Rule** or the **Trapezoidal Rule**, provide accurate solutions. The Python program used here demonstrates how numerical methods can be applied to compute solutions efficiently. First-order integral equations are crucial in modeling physical processes, economic systems, and other applications, and numerical methods are essential when analytical solutions are not feasible.

References:

1. Tikhonov, A. N., and Arsenin, V. Y., 'Solutions of Ill-Posed Problems,' Winston & Sons, 1977.
2. Kress, R., 'Linear Integral Equations,' Springer, 1999.
3. Groetsch, C. W., 'The Theory of Tikhonov Regularization for Fredholm Equations,' Springer, 1984.
4. Hansen, P. C., 'Rank-Deficient and Discrete Ill-Posed Problems,' SIAM, 1998.