

CYRR 304
Homework 9, Spring 2024

Name:

Homework 9 has questions 1 through 3 with a total of 20 points, with an opportunity to earn ten bonus points. The point value for each question or part of a question is in the box following each question or part of a question. This work is due **Saturday 13 April** at 11:59 PM.

An equation of the form

$$F(x) = g(x) + \int_0^1 K(x, y)F(y) dy$$

is a *Fredholm integral equation of the second kind*. The function F is the unknown and the functions g and K are given. The function K is known as the *kernel* of the equation.

Such equations have a rich history involving luminaries such as Norbert Wiener, Subrahmanyan Chandrasekhar, and John von Neumann. These equations have many applications, including scattering of light by planetary atmospheres, nuclear reactors, and digital filters.

Assuming both the functions g and K are continuous and that $\text{range}(K) \subset (-1, 1)$, the equation $F(x) = g(x) + \int_0^1 K(x, y)F(y) dy$ has a unique solution. Although there are methods for finding solutions to these equations, for most kernels K , the solution doesn't have a representation in terms of functions that we know anything about, so we need to resort to a numerical method. Our method will be to replace the integral by the trapezoidal rule; this gives

$$\begin{aligned} F(x) &= g(x) + \int_0^1 K(x, y)F(y) dy, \\ &\approx g(x) + \sum_{j=0}^n w_j K(x, x_j)F(x_j). \end{aligned}$$

where the weights w_0, w_1, \dots, w_n and the knots x_0, x_1, \dots, x_n are the appropriate values for the trapezoidal rule. Our next task is to find equations for F evaluated at each knot. To do that, we replace x by x_i in $F(x) \approx g(x) + \sum_{j=0}^n w_j K(x, x_j)F(x_j)$. That gives

$$F(x_i) = g(x_i) + \sum_{j=0}^n w_j K(x_i, x_j)F(x_j).$$

These are linear equations for the value of F at each knot. Here is a simple Julia language implementation of this method.

```
function integral_equation(g::Function, K::Function, n::Integer)
    w = k -> if k==0 || k==n 1/(2*n) else 1/n end
    M = [(if i==j 1 else 0 end) - w(j)*K(i/n, j/n) for i=0:n, j=0:n]
    b = [g(i/n) for i = 0:n]
    sol = M \ b
    x -> g(x) + sum([w(k) * K(x, k/n) * sol[k+1] for k = 0:n])
end
```

This code returns a Julia function that is our approximate solution.

- 10 1. The exact solution to $F(x) = x + \int_0^1 \frac{1}{2}F(y) dy$ is $F(x) = x + \frac{1}{2}$. Use the given code to numerically solve the equation

$$F(x) = x + \int_0^1 \frac{1}{2}F(y) dy. \quad (1)$$

You'll need to define $g = x \mapsto x$ and $K = (x, y) \mapsto \frac{1}{2}$. Graphically verify that the approximate solution and the true solution are identical. Use 10 panels to do this.

- 10 2. Use the given code to numerically solve the equation

$$F(x) = \frac{(2\pi^2 + 2) \sin(\pi x) - \pi}{2\pi^2} + \int_0^1 \frac{1}{2}|x - y|F(y) dy. \quad (2)$$

The exact solution to this equation is $F(x) = \sin(\pi x)$. Using $n = 50$, use Gadfly to graph the absolute value of the difference between the true solution and the numerically determined approximate solution. Visually determine the largest difference between the true solution and the numerically determined approximate solution.

3. (ten point bonus) Let $T_n(F)$ be the n -panel trapezoidal rule for a function F . The quadrature rule $\int_a^b F \approx \frac{4}{3}T_{2n}(F) - \frac{1}{3}T_n(F)$ is the well-known Simpson rule. The weights for this rule are

$$w_k = h \begin{cases} \frac{1}{3} & k = 0 \text{ or } k = n \\ \frac{4}{3} & k \text{ is odd} \\ \frac{2}{3} & k \text{ is even} \end{cases}, \quad (3)$$

where h is the step size. To modify our Julia code to use the Simpson rule, you'll need to replace

```
function integral_equation(g::Function, K::Function, n::Integer)
    w = k -> if k==0 || k==n 1/(2*n) else 1/n end
    M = [(if i==j 1 else 0 end) - w(j)*K(i/n,j/n) for i=0:n, j=0:n]
    b = [g(i/n) for i = 0:n]
    sol = M \ b
    x -> g(x) + sum([w(k) * K(x,k/n) * sol[k+1] for k = 0:n])
end
```

with

```
function integral_equation(g::Function, K::Function, n::Integer)
    w = k -> if k==0 || k==n 1/(3*n) elseif iseven(k) 2/(3*n) else 4/(3*n) end
    M = [(if i==j 1 else 0 end) - w(j)*K(i/n,j/n) for i=0:n, j=0:n]
    b = [g(i/n) for i = 0:n]
    sol = M \ b
    x -> g(x) + sum([w(k) * K(x,k/n) * sol[k+1] for k = 0:n])
end
```

Use the given code to numerically solve the equation

$$F(x) = \frac{(2\pi^2 + 2) \sin(\pi x) - \pi}{2\pi^2} + \int_0^1 \frac{1}{2} |x - y| F(y) \, dy. \quad (4)$$

using the Simpson rule method. Which method gives a more accurate value?