Integral Equations and Boundary Value Problems

Exercise, WS 2018/19

Programming Exercises 16.01.2019

Dr. Simon Hubmer, S2 503-1, simon.hubmer@ricam.oeaw.ac.at

Reminder: The programming exercises constitute 20% of your final grade!

This tutorial deals with the numerical implementation of the Nyström Method. Your code may be written only in `MATLAB`. Please send me your entire code per email until **16.01.2019** as one single **.zip** file with the name *FamilyName.zip*

Please put small comments in your code explaining the function of different code sections. This is an important requirement for all computer code and helps both me and you to read your code.

Consider the following integral equation:

$$\lambda x(s) - \int_0^1 k(s,t)x(t)\,dt = f(s)\,, \qquad s \in [0,1]\,.$$

Let us define the quadrature operator Q_n that approximates the integral of a continuous function,

$$\begin{aligned} Q_n &: C[0,1] \to \mathbb{R} \\ x &\mapsto \sum_{j=1}^{n} \omega_j x(t_j)\,. \end{aligned}$$

For the purpose of this tutorial, we will use Gaussian quadrature of order 4 on subintervals of $[0,1]$. The Gaussian quadrature of order 4 on the interval $[0,1]$ is given by four points $t_1,\dots,t_4$ and the corresponding weights $\omega_1,\dots,\omega_4$,

$$\begin{aligned} t_1 &= 0.06943184420297371 & \omega_1 &= 0.1739274225687269 \\ t_2 &= 0.33000947820757719 & \omega_2 &= 0.3260725774312731 \\ t_3 &= 1 - t_2 & \omega_3 &= \omega_2 \\ t_4 &= 1 - t_1 & \omega_4 &= \omega_1. \end{aligned}$$

Let $I_k := [\frac{k-1}{m}, \frac{k}{m}]$ for $k = 1,\dots,m$ be m equidistant subintervals of $[0,1]$. The Gaussian quadrature of order 4 on the interval I_k is given by four points t_j and corresponding weights ω_j correctly scaled to the interval I_k. By considering all subintervals I_k, we obtain $n = 4m$ points t_j and corresponding weights ω_j that define Q_n.

1. The interval $[0,1]$ is decomposed into m subintervals I_k. Gaussian quadrature of order 4 is used on each interval I_k. Write a function

$$[\texttt{t},\texttt{w}] = \texttt{gaussDomain(m)}\,,$$

that returns two vectors `t` and `w` of length `4m` that correspond to the quadrature points and weights on $[0,1]$ for evaluating integrals.

Test your code: e.g., $\int_0^1 t^2\,dt = 1/3$ should coincide with the numerical approximation `sum(w .* t.^2)`.

Quadrature points and weights t_j and ω_j define Q_n. The discretized integral operator K_n is given by $(K_n x)(s) := Q_n(k(s,\cdot)x)$ for any $s \in [0,1]$. By choosing $s = t_i$, we obtain the fully discretized problem $\lambda z - Mz = g$ where the matrix M and the vector g are given by

$$M_{ij} := \omega_j k(t_i, t_j), \qquad g_i := f(t_i)\,, \qquad i,j = \overline{1,n}\,.$$

2. Write functions

$$\texttt{M} = \texttt{assembleM(kernel,t,w)}\,,$$

and

$$\texttt{g} = \texttt{assembleg(func,t)}\,,$$

that create the matrix M and the vector g. Here `kernel` and `func` are functions with two and one argument respectively. Vectors `t` and `w` are generated by `gaussDomain`.

Vector z is computed by solving the linear system of equations $\lambda z - Mz = g$. Values of the solution x are obtained via z using the following interpolation:

$$x(s) := \frac{1}{\lambda}\left[\sum_{j=1}^{n} \omega_j k(s,t_j) z_j + f(s)\right]\,,$$

at any point $s \in [0,1]$.

3. Write a function

$$\texttt{x} = \texttt{interpNystrom(lambda,kernel,func,z,t,w,s)}\,,$$

that returns a vector `x` of interpolated values at points defined by the vector `s`. Variables `kernel`, `func`, `t` and `w` are as above; `z` is a vector of length n and `lambda` is a scalar parameter corresponding to λ.

4. Write a function

$$\texttt{x} = \texttt{solveNystrom}(\texttt{lambda}, \texttt{kernel}, \texttt{func}, \texttt{m}, \texttt{s})\,,$$

that solves the integral equation of the second kind numerically using the Nyström Method. Function `solveNystrom` should use all the functions defined above. You may use the `MATLAB` command $\texttt{z} = \texttt{A}\backslash\texttt{b}$ to solve a linear system of equations $\texttt{Az} = \texttt{b}$.

Variable `lambda` corresponds to λ, `kernel` to $k(\cdot,\cdot)$, and `func` to $f(\cdot)$; `m` is the number of subintervals used for discretization and the vector `s` defines the points where the solution vector `x` is to be computed.

5. Test your program. Try some of the equations you already solved:

(a)

$$x(s) - \int_0^1 (20st^2 + 12s^2t)x(t)\,dt = s\,, \qquad s \in [0,1]\,,$$

(b)

$$x(s) - \int_0^s x(t)\,dt = 1\,, \qquad s \in [0,1]\,,$$

(c)

$$x(s) - \int_0^s (t-s)x(t)\,dt = s\,, \qquad s \in [0,1]\,.$$

Define the functions `kernel` and `func`. Set $\texttt{s} = \texttt{0} : \texttt{0.01} : \texttt{1}$. Call your function: $\texttt{x} = \texttt{solveNystrom}(\texttt{lambda}, @\texttt{kernel}, @\texttt{func}, \texttt{m}, \texttt{s})$. Plot the solution: $\texttt{plot}(\texttt{s}, \texttt{x}, \texttt{s}, \texttt{sol})$, where `sol` is the vector of the function values of the real solution. E.g., $\texttt{sol} = \texttt{s/4} - \texttt{s.\^{}2/2}$ for the first equation (a) above.