Numerical Analysis Homework #9

due 2020 JUN 09, 9:50 a.m.

1 Assignments

Caution:

- To get full credit, *you must write down sufficient intermediate steps*, only giving the final answer earns you no credit!
- Please make sure that your handwriting is recognizable, otherwise you only get partial credit for the recognizable part.

I. Simpson's rule.

(a) Show that on $[-1, 1]$ Simpson's rule can be obtained as follows

$$\int_{-1}^{1} y(t)\mathrm{d}t = \int_{-1}^{1} p_3(y; -1, 0, 0, 1; t)\mathrm{d}t + E^S(y),$$

where $y \in \mathcal{C}^4[-1, 1]$ and $p_3(y; -1, 0, 0, 1; t)$ is the interpolation polynomial of y with interpolation conditions $p_3(-1) = y(-1)$, $p_3(0) = y(0)$, $p_3'(0) = y'(0)$, and $p_3(1) = y(1)$.

(b) Derive $E^S(y)$.

(c) Using (a), (b) and a change of variable, derive the composite Simpson's rule and prove the theorem on its error estimation.

II. Estimate the number of subintervals required to approximate $\int_0^1 e^{-x^2}\mathrm{d}x$ to 6 correct decimal places, i.e. the absolute error is no greater than 0.5×10^{-6},

(a) by the composite trapezoidal rule,

(b) by the composite Simpson's rule.

III. Gauss-Laguerre quadrature formula.

(a) Construct a polynomial $\pi_2(t) = t^2 + at + b$ that is orthogonal to $\mathbb{P}_1$ with respect to the weight function $\rho(t) = e^{-t}$, i.e.

$$\forall p \in \mathbb{P}_1, \qquad \int_0^{+\infty} p(t)\pi_2(t)\rho(t)\mathrm{d}t = 0.$$

(*hint*: $\int_0^{+\infty} t^m e^{-t}\mathrm{d}t = m!$)

(b) (10 points) Derive the two-point Gauss-Laguerre quadrature formula

$$\int_0^{+\infty} f(t)e^{-t}\mathrm{d}t = w_1 f(t_1) + w_2 f(t_2) + E_2(f)$$

and express $E_2(f)$ in terms of $f^{(4)}(\tau)$ for some $\tau > 0$.

(c) Apply the formula in (b) to approximate

$$I = \int_0^{+\infty} \frac{1}{1+t} e^{-t}\mathrm{d}t.$$

Use the remainder to estimate the error and compare your estimate with the true error. With the true error, identify the unknown quantity τ contained in $E_2(f)$.
(*hint*: use the exact value $I = 0.596347361\cdots$)

The above four problems weigh 15, 10, and 20 points, respectively.

2 `C++` programming

Write a `C++` function to perform discrete least square via QR factorization. Your algorithm should take as input the maximum degree of the fitting polynomial, three or more data pairs (x_i, y_i), and output coefficients of the fitted polynomial.

Run your subroutine on the following data.

x	0.0	0.5	1.0	1.5	2.0	2.5	3.0
y	2.9	2.7	4.8	5.3	7.1	7.6	7.7
x	3.5	4.0	4.5	5.0	5.5	6.0	6.5
y	7.6	9.4	9.0	9.6	10.0	10.2	9.7
x	7.0	7.5	8.0	8.5	9.0	9.5	10.0
y	8.3	8.4	9.0	8.3	6.6	6.7	4.1

In the notes, the condition number of a matrix A is defined as

$$\mathrm{cond}_A(\mathbf{x}) = \|A\|\|A^{-1}\|.$$

Report the condition number based on the 2-norm of the matrix G in the normal-equation approach (reuse results of your previous homework!) and that of the matrix R_1 in the QR-factorization approach, verifying that the former is much larger than the latter.

This programming assignment weighs 10 points. Thus the total number of points is 55 for this homework.

3 Extra credits

Additional 10% credits will be given to you if you typeset your solutions in LaTeX. You are welcome to use the LaTeX template available on my webpage. You can also get partial extra credit for typesetting solutions of *some* problems.

Note: If you choose to typeset your solutions in LaTeX, you still need to turn in a hard copy in class. In addition, please upload your latex source (.tex), supporting files, and `C++` program in a single zip file (**format**: `YourName_Homework9.zip`) to the course email `NumApproximation@163.com`.