Homework Problem Sheet 9

Introduction. This problem sheet is devoted to numerical integration.

Problem 9.1 Order of a Quadrature Rule

This simple problem provides some "calculus drill" in connection with fixing the weights of a quadrature formula to achieve the best possible order. Please refresh yourself the notion of "order" in connection with a quadrature rule and also study Lemma 4.1.A in the lecture notes NMI_4.

(9.1a) Let $f \in \mathcal{C}^0([-1,1])$ and $Q_w^{(2)}$ be a quadrature rule that approximates the weighted integral

$$I_w[f] := \int_{-1}^{1} f(x)w(x)\,\mathrm{d}x \approx Q_w^{(2)} := \alpha_0 f(-1) + \alpha_1 f(0) + \alpha_2 f(1) \tag{9.1.1}$$

with weight function $w(x) = 1 + x^2$. Compute the weights α_i, for $i = 0, 1, 2$ such that $Q_w^{(2)}$ has maximal order. What is this order?

HINT: Do not jump to conclusions! Remember from class that also the Simpson rule enjoys a "surprisingly high" order.

Problem 9.2 Gauss-Hermite Quadrature

Quadrature formulas can also be used to approximate the values of improper integrals. This problem discusses an example.

Let $f : \mathbb{R} \to \mathbb{R}$ be integrable, we define $I_w[f] := \int_{-\infty}^{\infty} f(t)w(t)\,\mathrm{d}t$ for the *weight function* $w(t) := \mathrm{e}^{-t^2}$. For $n \in \mathbb{N}$, let f_n denote the monomial $f_n(t) := t^n$.

(9.2a) From the Analysis course we know $I_w[f_0] = I_w[1] = \sqrt{\pi}$. Using integration by parts, prove that

$$I_w[t^n] = \begin{cases} 0 & n = 1, 3, \dots, \\ 2^{-n/2}\sqrt{\pi}(1 \cdot 3 \cdot 5 \cdot \ldots \cdot (n-1)) & n = 2, 4, \dots. \end{cases}$$

(9.2b) Determine a polynomial H_2 of degree 2 that satisfies $I_w[t^n H_2] = 0$ for $n = 0, 1$ and $I_w[H_2^2] = 1$, and $H_2(t) \overset{t\to\infty}{\longrightarrow} \infty$. This polynomial H_2 is called the *Hermite polynomial* of degree 2.

(9.2c) Find the zeros t_0, t_1 of H_2 with $t_0 < t_1$.

(9.2d) Let $\{\ell_0, \ell_1\}$ be the Lagrange interpolation polynomials associated to the points $\{t_0, t_1\}$ (the roots of H_2 from subproblem (9.2c)). Approximate the integral $I_w[f]$ for $f(t) = \cos(t)$ using the Gauss-Hermite quadrature formula given by

$$Q_w^{(1)}[f] := \sum_{i=0}^{1} f(x_i)\alpha_i \approx I_w[f],$$

where the weights are defined by $\alpha_i := I_w[\ell_i]$, $i = 0, 1$. Compare the result to the exact value

$$\int_{-\infty}^{\infty} \cos(t)\mathrm{e}^{-t^2}\,\mathrm{d}t = \frac{\sqrt{\pi}}{\mathrm{e}^{1/4}}.$$

Problem 9.3 Adaptive Quadrature

In Section 4.2 in NMI_4 we saw how to construct so-called composite quadrature rules based on partition (grid) of the integration interval. In class we did not discuss how to obtain such partitions and all examples used equidistant grids. However, often for a prescribed number of function evaluations a massive reduction of the quadrature error can be achieved by choosing a non-equidistant grid by taking into account features of the integrand. This can even be done automatically, based on *a posteriori error estimation*, as is demonstrated in this problem.

(9.3a) Implement a MATLAB-function

```
Qn = simpson(a,b,n,f)
```

that computes the integral $\int_a^b f(x)\,\mathrm{d}x$ using the composite Simpson rule $Q_{1/n}[f]$ on n intervals of the same size. The function takes as input the function handle `f` which, in turn, requires a single scalar argument.

(9.3b) Implement an *adaptive* composite Simpson rule

```
Qval = adaptiveSimpson_rec(a,b,f,tol)
```

according to the algorithm described below. It improves the grid adaptively, if the *estimated error* on an interval is greater than the (absolute) tolerance `tol`. The following strategy governs the refinement of the grid:

Adaptive grid refinement for composite Simpson rule: Assume you are given an interval $[a, b]$ of length $h = b - a$, the tolerance `tol` and the function f as a function handle.

i. *Estimate* the error of integration by using $err = |Q_h[f] - L_{h/2}[f]|$, where $Q_h[f]$ is the value produced by the Simpson rule and

$$L_{h/2}[f] = Q_h[f] + \frac{Q_{h/2}[f] - Q_h[f]}{15}$$

denotes the so-called extrapolated Simpson's rule.

ii. *Terminate* if $err \leq tol$. Then the approximation $L_{h/2}[f]$ is regarded as good enough and is returned as the approximate value `Qval` of the integral $\int_a^b f(x)\,\mathrm{d}x$.

iii. Otherwise *subdivide* $[a, b]$ into two subintervals of the same length and *recursively* call `adaptiveSimpson_rec` on the new intervals with half the value for the tolerance `tol`. The sum of the returned values is used as `Qval` then.

(9.3c) What is the *minimal* number of function evaluations required to compute the value returned by `adaptiveSimpson_rec(a,b,f,tol)` from sub-problem (9.3b), if it is known that $K \in \mathbb{N}_0$ recursive calls to this function have been made.

(9.3d) Implement a MATLAB function

```
Qval = adaptiveSimpson(a,b,fa,fm,fb,f,tol)
```

that, given the values $\texttt{fa} = f(a)$, $\texttt{fm} = f((a+b)/2)$ and $\texttt{fb} = f(b)$, computes the same approximate value for the integral $\int_a^b f(x)\,\mathrm{d}x$ as `adaptiveSimpson_rec(a,b,f,tol)` using the minimal number of f-evaluations found in subproblem (9.3c).

(9.3e) For the function $f(x) = 1/(10^{-4} + x^2)$ we compare the convergence of the equidistant composite Simpson rule and of the adaptive composite Simpson rule from subproblem (9.3d) on the interval $[0, 1]$. Create a doubly logarithmic plot of the quadrature error versus the number of point evaluations for both quadrature rules.

HINT:

- The exact value of the integral is $\int_0^1 f(x)\,\mathrm{d}x = 10^2\text{arctan}(10^2)$.
- The equidistant composite Simpson rule should be used with $1, \ldots, 1000$ grid intervals.
- The adaptive quadrature rule should be applied with the values 0.7^ℓ, $\ell = 1, \ldots, 100$ for the tolerance `tol`.
- To count the number of function evaluations in `adaptiveSimpson` use a *global variable*. Obtain information about the use of the `global` keyword from the MATLAB documentation.

Published on April 28, 2014.

To be submitted on May 6/7, 2014.

MATLAB: Submit all files in the online system. Include the files that generate the plots. Label all your plots. Include commands to run your functions. Comment on your results.

Written exercises: Hand-in the solutions during the exercise class or in the labeled boxes in HG G 53.x.

References

[NMI] Lecture Slides for the course "Numerical Analysis I".

Last modified on April 28, 2014