NumCSE

Autumn Semester 2017

Prof. Rima Alaifari

Exercise sheet 10
Numerical Quadrature

P. Bansal

Problem 10.1: Numerical integration of improper integrals

In this problem, we study two methods for computing improper integrals of the form $\int_{-\infty}^{\infty} f(t)dt$, where $f : \mathbb{R} \to \mathbb{R}$ is a continuous function that decays sufficiently fast for $|t| \to \infty$ (so that it is integrable on $\mathbb{R}$).

One option is to approximate the integral by truncating the domain to a bounded interval $[-b, b]$, $b \leq \infty$:

$$I = \int_{-\infty}^{\infty} f(t)dt \approx \int_{-b}^{b} f(t)dt.$$

Then, standard quadrature rules can be used to compute this integral over $[-b, b]$.

(a) Given the integrand $g(t) := 1/(1+t^2)$, find b such that the truncation error E_T satisfies:

$$E_T := \left| \int_{-\infty}^{\infty} g(t)dt - \int_{-b}^{b} g(t)dt \right| \leq \epsilon \tag{10.1}$$

for some $\epsilon > 0$.

(b) What is the challenge in using the truncation approach for a generic integrand?

Another option is to transform the improper integral I to a bounded domain by substitution. For instance, we may substitute $t = \cot(s)$.

(c) Rewrite the integral I using the substitution $t = \cot(s)$.

(d) Simplify the transformed integral from subproblem 10.1.(c) explicitly for $g(t) := \frac{1}{1+t^2}$.

Hint: Refer Pythagorean trigonometric identities.

(e) Implement a C++ function that uses the substitution $t = \cot(s)$ and the n-point Gauss-Legendre quadrature rule to evaluate $\int_{-\infty}^{\infty} f(t)dt$:

```
template <typename Function>
double quadinf(int n, const Function & f);
```

Use the function `golubwelsh` (see `golubwelsh.hpp`) to compute the nodes and weights of the Gauss quadrature rule.

(f) Study the convergence behaviour of the quadrature method, for $n \to \infty$, implemented in the previous subproblem for the integrand $h(t) := \exp(-(t-1)^2)$ (shifted Gaussian). For error computation, you can use that $\int_{-\infty}^{\infty} h(t)dt = \sqrt{\pi}$.

Problem 10.2: Smooth integrand by transformation

Knowledge about the structure of a non-smooth integrand can be used to restore its smoothness by transformation. We explore this idea in the current problem.

Let $f : [-1, 1] \to \mathbb{R}$ be a smooth, odd function. We want to approximate the integral:

$$I := \int_{-1}^{1} \arcsin(t)\, f(t)\, \mathrm{d}t \tag{10.2}$$

using a suitable quadrature rule. Note that $d(\arcsin(t))/dt = 1/\sqrt{1-t^2}$, which is not defined at $t = \pm 1$.

We use Gauss quadrature, however, its detailed knowledge is **not** important to solve this problem. A C++ function `gaussquad` (see `gaussquad.hpp`) to compute the nodes $\mathbf{x}$ and the weights $\mathbf{w}$ of an n-point Gaussian quadrature on $[-1, 1]$ is provided:

```
QuadRule gaussquad(const unsigned n);
```

Here, `QuadRule` is a `struct` containing the quadrature weights `w` and nodes `x`:

```
struct QuadRule {
  Eigen::VectorXd nodes, weights;
};
```

(a) Implement a C++ function:

```
template <class Function>
void nonSmoothIntegrand(const Function & f);
```

to study the convergence behaviour of the quadrature error versus the number of quadrature points $n = 1, \ldots, 50$. Here `f` is a handle to the function f.

(b) Plot the convergence for $f(t) = \sinh(t)$ and describe it qualitatively and quantitatively. Use $I \approx 0.870267525725852642$ as the exact value of the integral.

(c) Transform the integral I using a suitable variable substitution, so that the integrand becomes smooth, if the given $f \in C^\infty([-1, 1])$ is smooth.

(d) Implement a C++ function:

```
template <class Function>
void smoothIntegrand(const Function & f);
```

to study the convergence behaviour of the quadrature error versus the number of quadrature points $n = 1, \ldots, 50$, as in Sub-problem (a).

(e) Plot the convergence behaviour for $f(t) = \sinh(t)$ using `smoothIntegrand` and compare with the plot obtained from Sub-problem (b).