Prof. Ch. Schwab
B. Fitzpatrick
J. Zech

Numerische Mathematik I

ETH Zürich
D-MATH

Homework Problem Sheet 9

Problem 9.1 Composite Trapezoidal Rule with Non-Uniform Grid

By using a grid with a non-uniform stepsize it may be possible to achieve a higher quality approximation for integrals where the integrand is smooth but problematic at certain points. Let $f(x) = \sqrt{x}$ and note that it is not continuously differentiable in $[0, 1]$. We first consider a grid with uniform stepsize $h = N^{-1}$.

(9.1a) Show that the following error estimate holds

$$\left| \int_0^1 f(x)\,\mathrm{d}x - Q^{(1)}_{[0,1]}[f] \right| \leqslant Ch^{\frac{3}{2}},$$

where $Q^{(1)}_{[0,1]}[\cdot]$ denotes the composite Trapezoidal rule.

Solution: We have

$$I_{[0,1]}[f] = \sum_{j=1}^{N} I_{[x_{j-1},x_j]}[f] = I_{[0,\frac{1}{N}]}[f] + \sum_{j=2}^{N} I_{[x_{j-1},x_j]}[f].$$

Define $\mathcal{I}^{(1)} = I_{[0,\frac{1}{N}]}[\sqrt{x}]$ and $\mathcal{I}^{(2)} = \sum_{j=2}^{N} I_{[x_{j-1},x_j]}[\sqrt{x}]$.

First we estimate the error due to $\mathcal{I}^{(2)}$. For each of the subintervals in $\mathcal{I}^{(2)}$ we have that $\sqrt{x} \in C^2([x_{j-1}, x_j])$. Therefore, applying Theorem 4.2, for $j = 2, \ldots, N$ we have that for some $\xi \in [x_{j-1}, x_j]$

where we have used the fact that the maximum of $|f''(\xi)|$ occurs at the left endpoint for $\xi \in [x_{j-1}, x_j]$ due to the monotonicity of $|f''(\xi)| = \left|\frac{1}{4}\xi^{-\frac{3}{2}}\right|$. Summing the contribution to the error from each of the subintervals we have

$$\left|\sum_{j=2}^{N}(I_{[x_{j-1},x_j]}[f] - Q^{(1)}_{[x_{j-1},x_j]}[f])\right| \leq \sum_{j=2}^{N}\left|I_{[x_{j-1},x_j]}[f] - Q^{(1)}_{[x_{j-1},x_j]}[f]\right|$$

$$\leq$$

Regarding the $\mathcal{I}^{(1)}$ term, $\sqrt{x}$ is not differentiable at $x = 0$ and so $\sqrt{x} \notin C^2([0,h])$. As we can't use the estimate in Theorem 4.2, we calculate the error directly as follows

$$\left| I_{[0,h]}[f] - Q^{(1)}_{[0,h]}[f] \right| = \left| \int_0^h \sqrt{x} dx - Q_{[0,h]}[f] \right| =$$

Summing $\mathcal{I}^{(1)}$ and $\mathcal{I}^{(2)}$ we obtain the upper bound error estimate of $Ch^{\frac{3}{2}}$.

(9.1b) Letting $x_0 = 0$, propose a selection of $\{x_j\}_1^N$ such that

$$\left| I_{[x_{j-1},x_j]}[f] - Q^{(1)}_{[x_{j-1},x_j]}[f] \right| \leqslant CN^{-2}.$$

HINT: Choose $x_j = (\frac{j}{N})^\beta$ for some $\beta > 1$ to equilibrate all

$$\left| I_{[x_{j-1},x_j]}[f] - Q^{(1)}_{[x_{j-1},x_j]}[f] \right|,$$

such that the undesirable effect on the error estimate due to $\sqrt{x}$ not being differentiable at the left endpoint of the first subinterval is compensated for.

Solution:

$$I_{[0,1]}[f] = I_{[0,x_1]}[f] + \sum_{j=2}^{N} I_{[x_{j-1},x_j]}[f].$$

Define $\mathcal{I}^{(1)} = I_{[0,x_1]}[\sqrt{x}]$ and $\mathcal{I}^{(2)} = \sum_{j=2}^N I_{[x_{j-1},x_j]}[\sqrt{x}]$. For each of the subintervals in $\mathcal{I}^{(2)}$ we have that $\sqrt{x} \in C^2([x_{j-1}, x_j])$. Therefore, applying Theorem 4.2, for $j = 2, \dots, N$ we have that

Let $x_j = (\frac{j}{N})^\beta$ and $g(x) = (\frac{x}{N})^\beta$. Taking the derivative of $g(x)$ gives

$$g'(x) = \beta\left(\frac{x}{N}\right)^{\beta-1} \cdot \frac{1}{N}$$

Therefore

$$|x_j - x_{j-1}| =$$

So taking the sum over all the subintervals we have

$$\left|\sum_{j=2}^{N}(I_{[x_{j-1},x_j]}[f] - Q^{(1)}_{[x_{j-1},x_j]}[f])\right| \leqslant \sum_{j=2}^{N}\left|I_{[x_{j-1},x_j]}[f] - Q^{(1)}_{[x_{j-1},x_j]}[f]\right| = \sum_{j=2}^{N}\left|-\frac{(x_j - x_{j-1})^3}{12}f''(\xi)\right|$$
$$\leq$$

Note that $\sup_{j\geqslant 2}(\frac{j-1}{j})^{-\frac{3}{2}} \leqslant 2^{\frac{3}{2}}$ as $\frac{j-1}{j} \in [\frac{1}{2}, 1]\ \forall\, j \geqslant 2$. Therefore the previous expression is

Again, noting the issue at the left endpoint of the first subinterval, we calculate the error for the $\mathcal{I}^{(1)}$ term directly

Summing $\mathcal{I}^{(1)}$ and $\mathcal{I}^{(2)}$ we obtain the upper bound error estimate of CN^{-2}.

Problem 9.2 Quadrature Rule Based on Interpolation

Let $f \in \mathcal{C}(\mathbb{R})$ be bounded. We want to calculate $\mathsf{I}_{[-1,1]}[f] = \int_{-1}^{1} f(x)\,\mathrm{d}x$ using a quadrature rule based on interpolation.

(9.2a) Define a quadrature rule based on interpolation $Q^{x_0}_{[-1,1]}[f] = \sum_{i=0}^{2} w_i f(x_i)$ with the sampling points $x_1 = -1$ as well as $x_2 = 1$ and the variable parameter $x_0 < -1$, which integrates all polynomials $p \in \mathbb{P}_n$ with $n \in \mathbb{N}$ as big as possible exactly. In order to do this deduce the weights

$$w_0(x_0) = -\frac{4}{3(-1+x_0)(1+x_0)}, \quad w_1(x_0) = \frac{1+3x_0}{3(1+x_0)}, \quad w_2(x_0) = \frac{-1+3x_0}{3(-1+x_0)}.$$

What quadrature rule do you get? What is the degree of your quadrature rule? What is the sign of the weights for $x_0 < -1$?

Solution: The rule is of form $Q[f] = w_0(x_0)f(x_0) + w_1(x_0)f(x_1) + w_2(x_0)f(x_2)$. The weights can be determined by solving the following equations

$$w_0 + w_1 + w_2 = \int_{-1}^{1} 1 \,\mathrm{d}x = 2$$

By using Gaussian elimination and back-substitution we can calculate the results given in the task.

(9.2b) Determine the condition of the quadrature rule depending on x_0. How does the condition behave for $x_0 \to -\infty$, resp. $x_0 \nearrow -1$?

Solution: The condition of the quadrature rule is the smallest number $\kappa > 0$, for which the following holds

$$|Q[f] - Q[g]| \leq \kappa \|f - g\|_\infty \quad \forall\, f, g \in \mathcal{C}^0(\mathbb{R}).$$

We observe:

$$|Q[f] - Q[g]| = |w_0(f(x_0) - g(x_0)) + w_1(f(x_1) - g(x_1)) + w_2(f(x_2) - g(x_2))| \qquad \leq$$

therefore $\kappa = |w_0| + |w_1| + |w_2|$. In this case we have

For $x_0 \to -\infty$ we observe the following behaviour

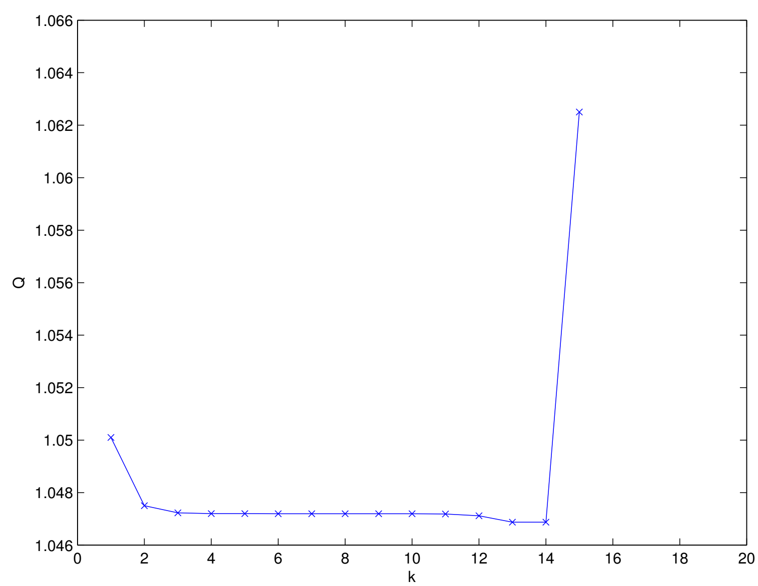
For $x_0 \nearrow -1$ we observe the following behaviour

(9.2c) Estimate the roundoff error when using $Q^{x_0}_{[-1,1]}[f]$ in floating-point arithmetic $\mathbb{F}(10, 6, -10, 10)$, if $x_0 = -1 - 10^{-6}$. Assume that we can evaluate f exactly up to machine precision.

Solution: In a finite floating-point arithmetic we do not evaluate the exact function f, but an approximation $\widetilde{f}$. If we assume that we can evaluate f exactly up to machine precision in $\mathbb{F}(\beta, t, e_{\min}, e_{\max})$, we have $\|f - \widetilde{f}\|_\infty \leq \frac{1}{2}\beta^{1-t}$ with the Unit Roundoff $\frac{1}{2}\beta^{1-t}$. Using the condition we get an estimate of the roundoff error

(9.2d) Write a MATLAB-script `quadrature.m`, that calculates $Q^{x_0}_{[-1,1]}[f]$ for $f(x) = \frac{\pi}{4}\cos(\frac{\pi}{4}x + \frac{\pi}{4})$, where $x_0 = -1 - 10^{-k}$, $k = 1, 2, \ldots, 20$. Display the behaviour of $Q^{x_0}_{[-1,1]}[f]$ graphically depending on k and explain the result.

Solution:



(9.2e) Let $x_0 = -10^k$, $k = 2, 3, \ldots$. Determine the quadrature rule $Q_{[-1,1]}^{x_0 \to -\infty}[f]$ for the limit $k \to \infty$.

Solution:

Problem 9.3 Romberg Extrapolation

(9.3a) Implement a MATLAB function `RombergExtr(f,a,b,m)` which computes the extrapolation table associated with Romberg scheme for $I[f] = \int_a^b f(x)\,dx$ as in [NMI, Sect. 4.3.3] and with $h = (b-a)/2$.

HINT: Use the routine `trapez.m` in [NMI, Sect. 4.2].

(9.3b) Implement a MATLAB function

`ErrConv(f,a,b,m,Iref)`

to compute the error of the Romberg extrapolation applied to $I[f] = \int_a^b f(x)\,dx$ with respect to the analytical value `Iref` of the integral $I[f]$. Use the implementation derived in (9.3a). For the ith-column of the Romberg table ($i = 0, \ldots, m$), plot the error vs $H = 2^{-i}h$ and determine *numerically* the convergence order. Explain the results you obtain for $f = \sin(x)$, $a = 0, b = \pi$ and $m = 10$.

HINT: Compare with the plots in [NMI, Sect. 4.2].

(9.3c) Repeat the numerical analysis performed in (9.3b) for $I[f] = \int_0^1 \sqrt{x}\,dx$. What can you observe? Compare with the results in (9.3b).

Problem 9.4 Gauss Quadrature rule for non constant weight

Determine the unique Gauss quadrature formula that has points $x_0, x_1 \in [-1, 1]$ and weights $A, B \in \mathbb{R}$ for the quadrature rule

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)\,\mathrm{d}x \approx Af(x_0) + Bf(x_1)$$

such that the formula has the highest possible degree. What is that degree?

HINT: The integrand is invariant under $x \to -x$ so we look for symmetric points x_0, x_1.

Problem 9.5 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.

(9.5a) 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.

Solution:

(9.5b) 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.

Solution:

(9.5c) What is the *minimal* number of function evaluations required to compute the value returned by `adaptiveSimpson_rec(a,b,f,tol)` from sub-problem (9.5b), if it is known that $K \in \mathbb{N}_0$ recursive calls to this function have been made.

Solution:

(9.5d) 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 (9.5c).

Solution:

(9.5e) 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 (9.5d) 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 $\int_0^1 f(x)\,\mathrm{d}x = 10^2 \arctan(10^2)$.
- The equidistant composite Simpson rule should be used with $1,\dots,1000$ grid intervals.
- The adaptive quadrature rule should be applied with the values 0.7^ℓ, $\ell = 1,\dots,100$ 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.

Solution:

Listing 9.1: Solution for subproblem (9.5e)

```
clear all

global count

f = @(x) 1./(1e-4 + x.^2);
a = 0; b = 1;
```

```
neq = 1000; nad = 100;
% Exact integral value
Qex = 100*atan(100);

% Equidistant composite Simpson rule
% TODO

% Adaptive composite Simpson rule
% TODO

figure()
loglog(ceq, err, '--k', cad, errAd, '-r');
legend('Equidistant Simpson', 'Adaptive Simpson');
xlabel('Number of function evaluations');
ylabel('Error of Quadrature');
```

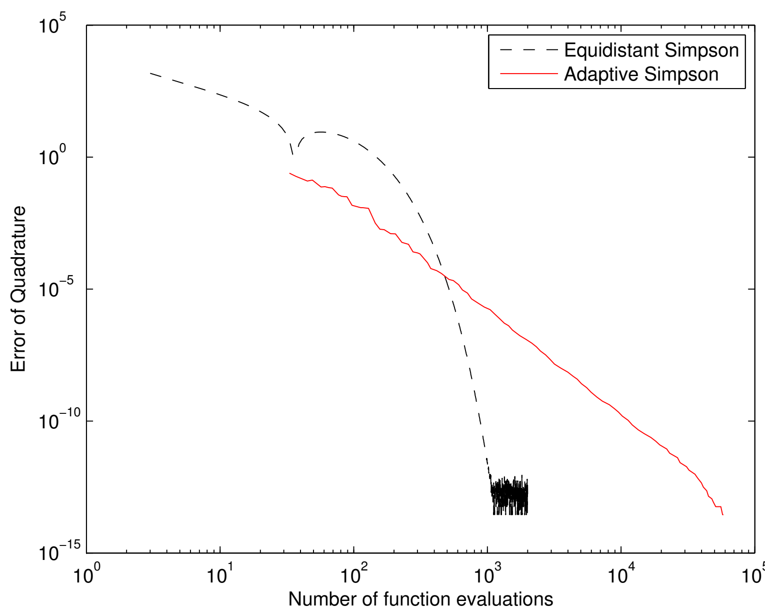


Figure 9.1: Error of quadrature

The adaptive quadrature formula reaches a smaller error on the same amount of subintervals. Since the number of subintervals is proportional to the number of evaluations of the function, we can call the adaptive quadrature more efficient (the overhead costs induced by the error estimation and the recursion are neglected).

Published on April 28, 2015.

To be submitted on May 10, 2015.

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.

References

[NMI] Lecture Notes for the course “Numerische Mathematik I”.

Last modified on May 2, 2016