

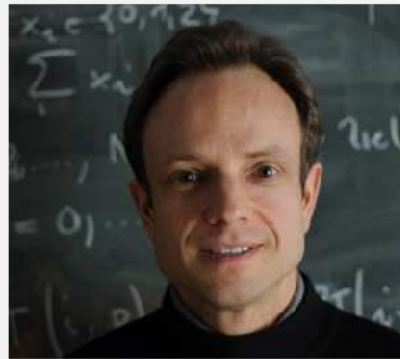
Course Video

Section 7.6: Adaptive Quadrature

Prof. R. Hiptmair, SAM, ETH Zurich

Date: November 22, 2020

(C) Seminar für Angewandte Mathematik, ETH Zürich



Dependencies. Section 7.5

Duration: 20 minutes

VII Numerical Quadrature

7.6. Adaptive Quadrature

General construction of composite quadrature rules

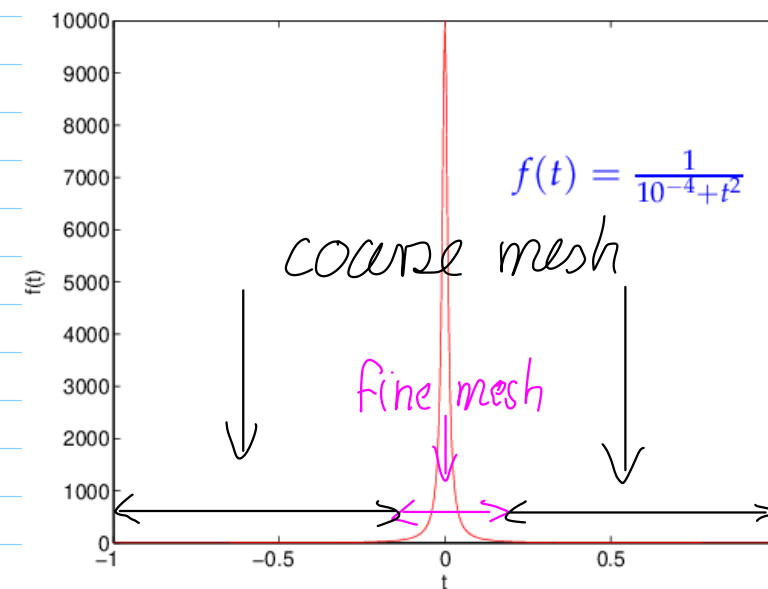


Idea: Partition integration domain $[a, b]$ by a **mesh**/grid ($\rightarrow$ Section 6.6)
 $\mathcal{M} := \{a = x_0 < x_1 < \dots < x_m = b\}$

Apply quadrature formulas from Section 7.3, Section 7.4 **locally** on mesh intervals $I_j := [x_{j-1}, x_j]$, $j = 1, \dots, m$, and sum up.

composite quadrature rule

Ex. 7.6.0.2 (Rationale for adaptive quadrature)



Integrate by composite trapezoidal rule

Goal: Automated choice of good mesh

(2)

§ 7.6.0.4 (Goal: Equidistribution of errors)

Ex: composite T.R., local quad. error

$$\left| \int_{x_{k-1}}^{x_k} f(t) dt - \frac{1}{2}(f(x_{k-1}) + f(x_k)) h_k \right| \leq \frac{1}{8} h_k^3 \|f''\|_{L^\infty([x_{k-1}, x_k])}, \quad h_k := x_k - x_{k-1}. \quad (7.6.0.3)$$

Heuristics: choosing mesh such that local errors are **equidistribution**

$$h_k : h_k^3 \|f''\|_{L^\infty([x_{k-1}, x_k])} = C \quad \forall k$$

A priori:

based on prior knowledge about f

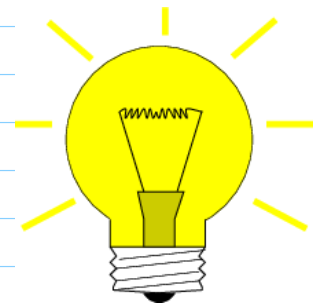
A posteriori:

"automatic" based on information gained during the computation.

▷ A posteriori adaptive mesh refinement

Adaptation loop for numerical quadrature

- ➔ (1) **ESTIMATE**: based on available information compute an approximation for the quadrature error on every mesh interval.
- (2) **CHECK TERMINATION**: if total error sufficient small → **STOP**
- (3) **MARK**: single out mesh intervals with the largest or above average error contributions.
- ➔ (4) **REFINE**: add node(s) inside the marked mesh intervals. GOTO (1)



Idea: local error estimation by comparing local results of two quadrature formulas Q_1, Q_2 of different order → local error estimates

heuristics: $\text{error}(Q_2) \ll \text{error}(Q_1) \Rightarrow \text{error}(Q_1) \approx Q_2(f) - Q_1(f)$.

Here: Q_1 = trapezoidal rule (order 2) ↔ Q_2 = Simpson rule (order 4)

Details: Initial mesh $\mathcal{M} = \{x_j\}_{j=0}^m$

❶ (Error estimation)

For $I_k = [x_{k-1}, x_k], k = 1, \dots, m$ (midpoints $p_k := \frac{1}{2}(x_{k-1} + x_k)$)

$$\text{EST}_k := \left| \underbrace{\frac{h_k}{6}(f(x_{k-1}) + 4f(p_k) + f(x_k))}_{\text{Simpson rule}} - \underbrace{\frac{h_k}{4}(f(x_{k-1}) + 2f(p_k) + f(x_k))}_{\text{trapezoidal rule on split mesh interval}} \right|. \quad (7.6.0.11)$$

order = 4

order = 2

③

② (Check termination)

Simpson rule on $\mathcal{M} \Rightarrow$ intermediate approximation $I \approx \int_a^b f(t) dt$ If $\sum_{k=1}^m \text{EST}_k \leq \text{RTOL} \cdot I$ ($\text{RTOL} :=$ prescribed relative tolerance) $\Rightarrow$ **STOP***

(7.6.0.12)

 $\uparrow$ estimate for total error of trapezoidal rule

* return approximation by Simpson rule

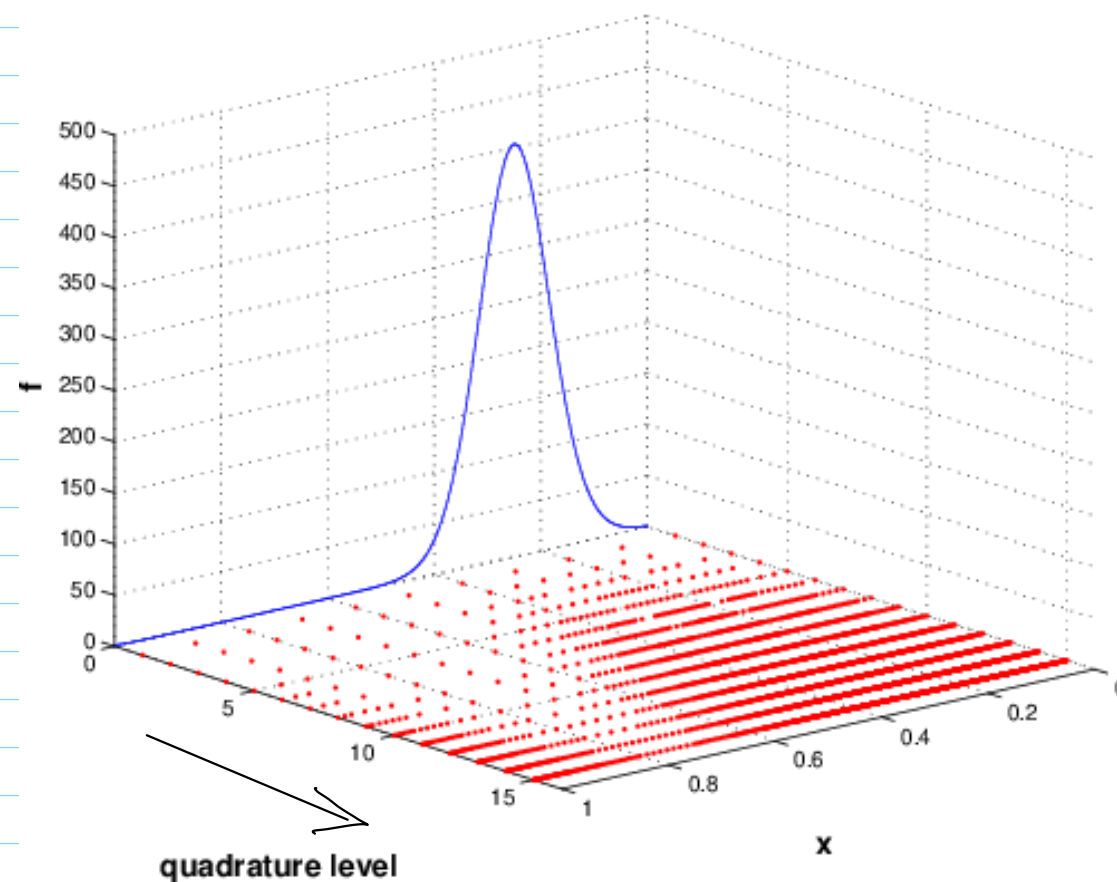
③ (Marking)

Marked intervals: $\mathcal{S} := \{k \in \{1, \dots, m\} : \text{EST}_k \geq \eta \cdot \underbrace{\frac{1}{m} \sum_{j=1}^m \text{EST}_j}_{\text{average local error}}\}$, $\eta \approx 0.9$. (7.6.0.13)

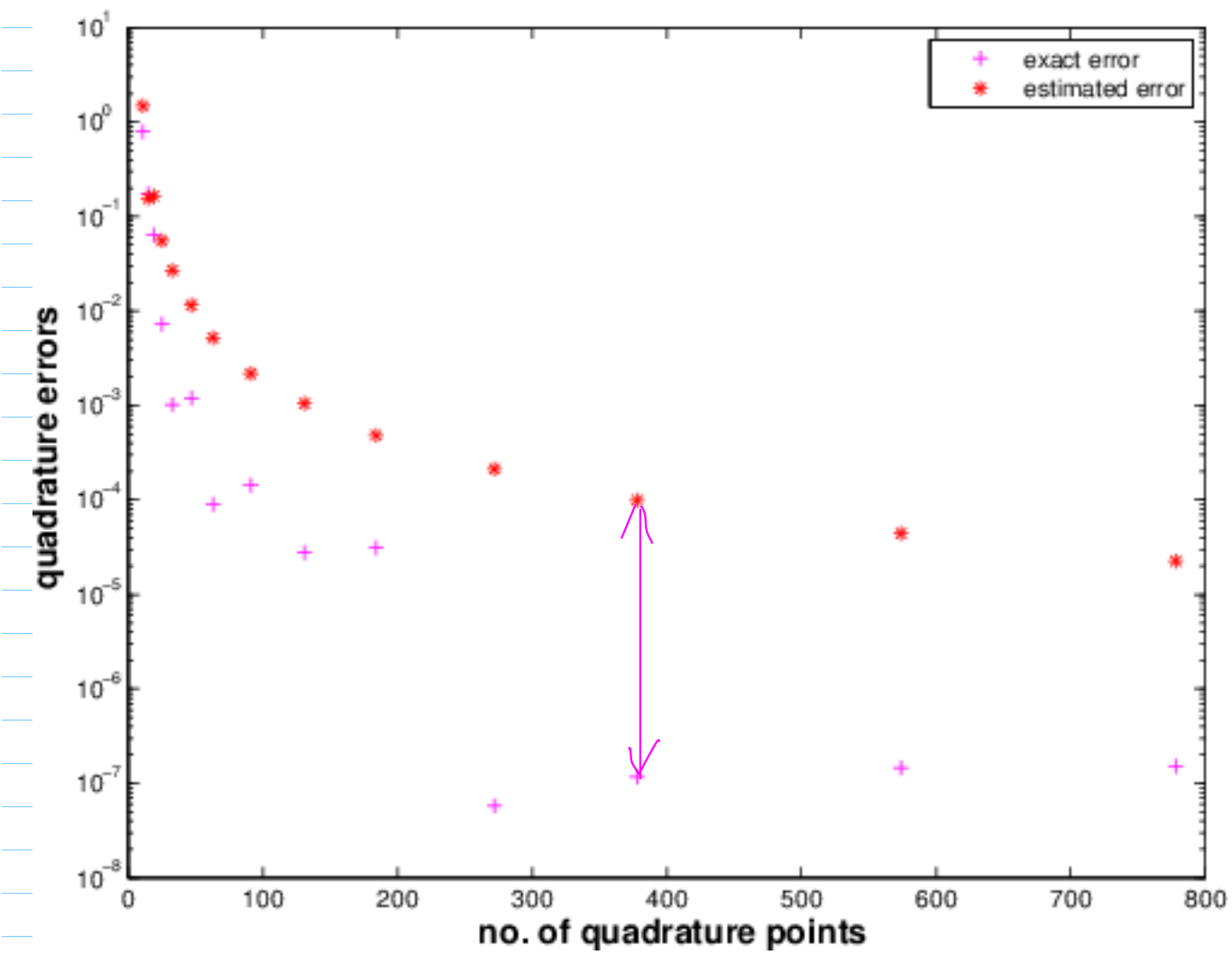
④ (Local mesh refinement)

new mesh: $\mathcal{M}^* := \mathcal{M} \cup \{p_k := \frac{1}{2}(x_{k-1} + x_k) : k \in \mathcal{S}\}$. (7.6.0.14)Then continue with step ① and mesh $\mathcal{M} \leftarrow \mathcal{M}^*$. $\uparrow$
add midpoints of marked cells $\Downarrow$
 $\Rightarrow$ adaptive loop

Exp. 7.6.0.18 (h-adaptive numerical quadrature)

 $f(t) = \exp(6 \sin(2\pi t))$ on $[0, 1]$ * = quadrature nodes
"reasonably distributed"

4



▷ EST_K far off,
but refinement control work

