TDT4127 Programming and Numerics Week 45

Adaptive Simpson's method

- A recursive look at integration

Learning goals

- Goals
  - Computing integrals
  - **Adaptive Simpson's method**
  - Recursion

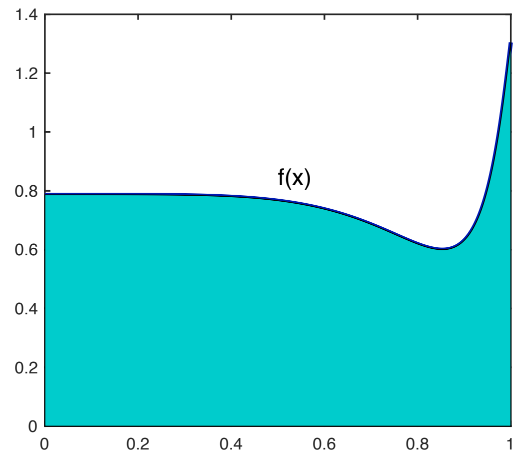
- Curriculum
  - Exercise set 10
    - Note: This set counts as two exercises
      - **If you do the chess exercise**

Numerical integration - repetition

- Everyone **loves** to integrate! But it can be hard.

$$\int_a^b f(x)\mathrm{d}x = ?$$

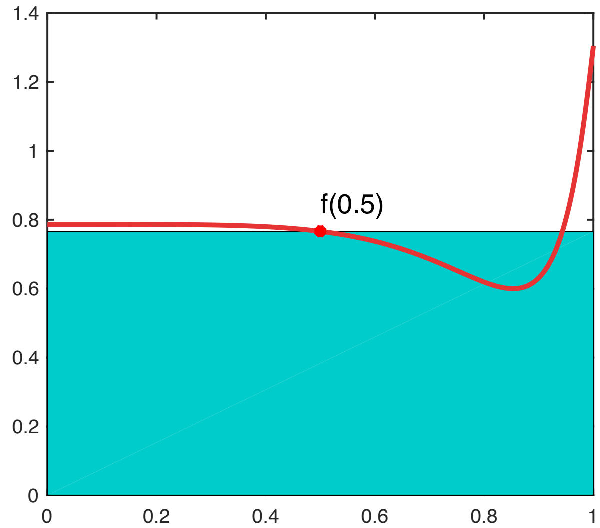
- Integrating in 1D ⇔ Finding area under the graph



- The idea: **Approximate $f(x)$ by something easier**
  - ***Polynomials*** are really easy and approximate well!

Midpoint rule - repetition

- Approximate f by a **constant**, $f\left(\frac{a+b}{2}\right)$, and integrate:

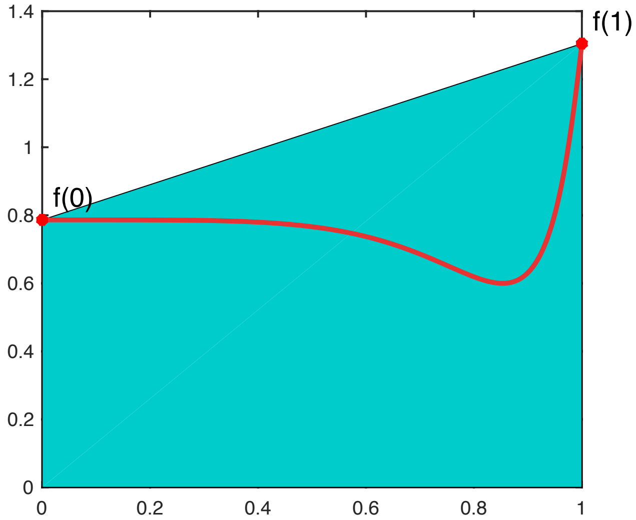


$$\int_a^b f(x)\mathrm{d}x \approx f\left(\frac{a+b}{2}\right)(b-a)$$

Trapezoidal rule - repetition

- Approximate f by a **linear** polynomial g and integrate:

$$g(x) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a}$$

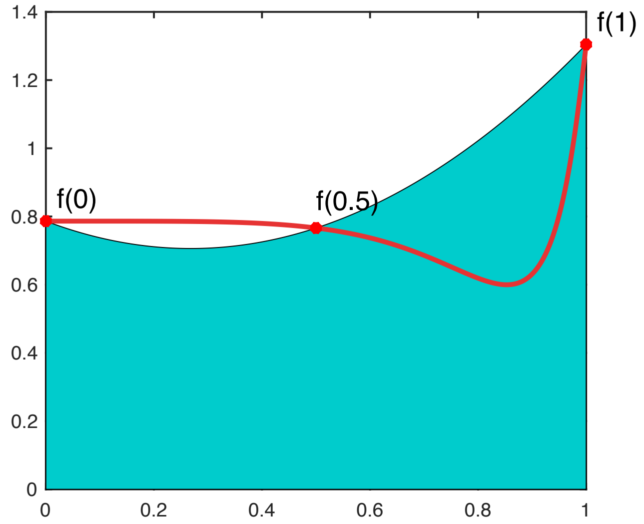


$$\int_a^b f(x)\mathrm{d}x \approx \big(f(a) + f(b)\big)\frac{b-a}{2}$$

Simpson's rule - repetition

- Approximate f by a **quadratic** polynomial g and integrate:

$$g(x) = f(a)\frac{(x-b)(x-c)}{(a-b)(a-c)} + f(b)\frac{(x-a)(x-c)}{(b-a)(b-c)} + f(c)\frac{(x-a)(x-b)}{(c-a)(c-b)}$$



$$\int_a^b f(x)\mathrm{d}x \approx \left(f(a) + 4f(c) + f(b)\right)\frac{b-a}{6}$$

Composite rules - repetition

- Split [*a*,*b*] into smaller ***subintervals***, *approximate the integrals*

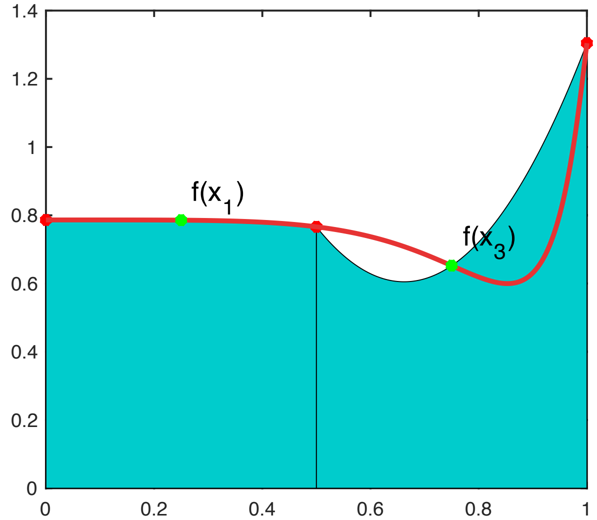
$$\int_a^b f(x)\mathrm{d}x = \int_a^c f(x)\mathrm{d}x + \int_c^b f(x)\mathrm{d}x \approx \int_a^c g(x)\mathrm{d}x + \int_c^b g(x)\mathrm{d}x$$

- This is called a ***composite*** method
- We called the number of subintervals N
- We considered subintervals of fixed width h
- Splitting an interval of width (b-a) into N parts gives h=(b-a)/N.

Composite Simpson's rule

- Use a **quadratic** approximation on each subinterval
  - Subintervals: $[x_{2k}, x_{2k+2}],\ \ k = 0, \ldots, N-1,$

$$x_k = a + kh, \qquad h = \frac{b-a}{2N}$$

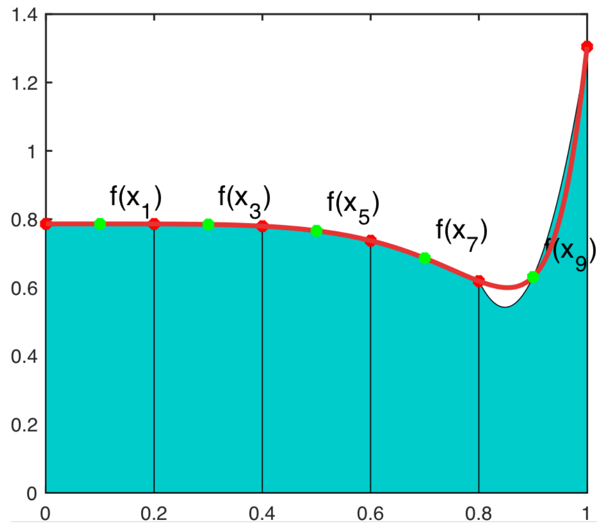


$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{2N-2}) + 4f(x_{2N-1}) + f(x_{2N})\right)$$

Composite Simpson's rule

- Use a **quadratic** approximation on each subinterval
  - Subintervals: $[x_{2k}, x_{2k+2}]$, $k = 0, \dots, N-1$,

$$x_k = a + kh, \qquad h = \frac{b-a}{2N}$$

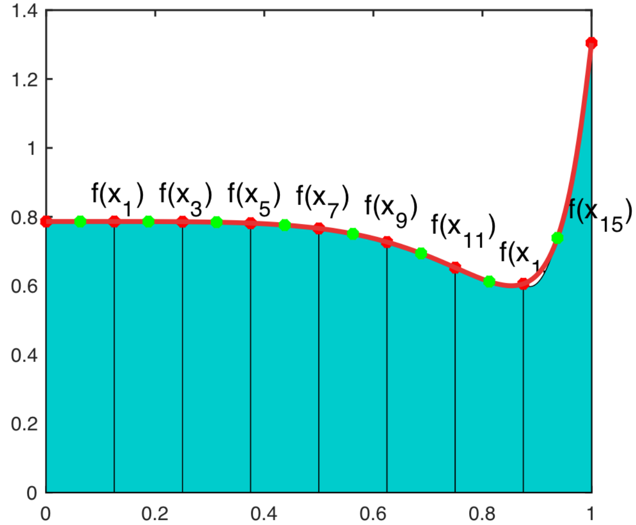


$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{2N-2}) + 4f(x_{2N-1}) + f(x_{2N})\right)$$

Composite Simpson's rule

- Use a **quadratic** approximation on each subinterval
  - Subintervals: $[x_{2k}, x_{2k+2}], \quad k = 0, \dots, N-1,$

$$x_k = a + kh, \qquad h = \frac{b-a}{2N}$$



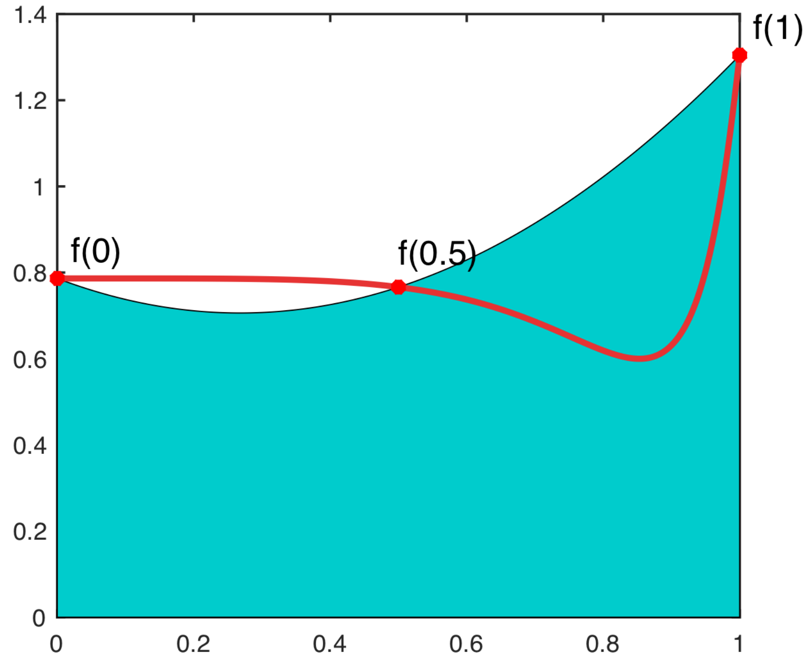
$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{3}\big(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{2N-2}) + 4f(x_{2N-1}) + f(x_{2N})\big)$$

Inefficiency in composite rules

- *Criticism*: We add more points ***even in areas that don't need better approximations***
  - From the previous example, see the left half of the interval
- Problem: more function evaluations than necessary
  - More function evaluations $\Rightarrow$ longer running time
- We want to **improve** the **efficiency** by only splitting into more subintervals **where necessary**

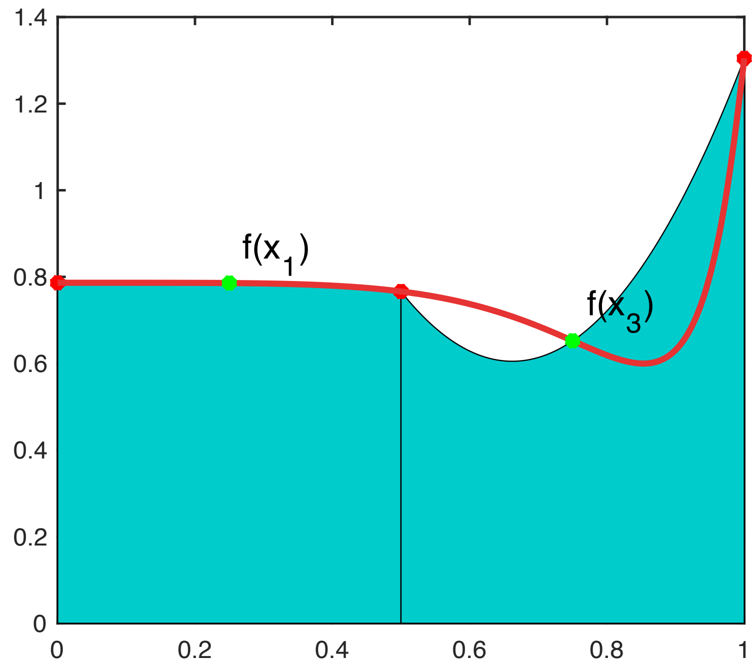
Adaptive refinement example

- The integral is **not good enough**, *needs refinement*
- The **whole** interval should be **split** in more subintervals



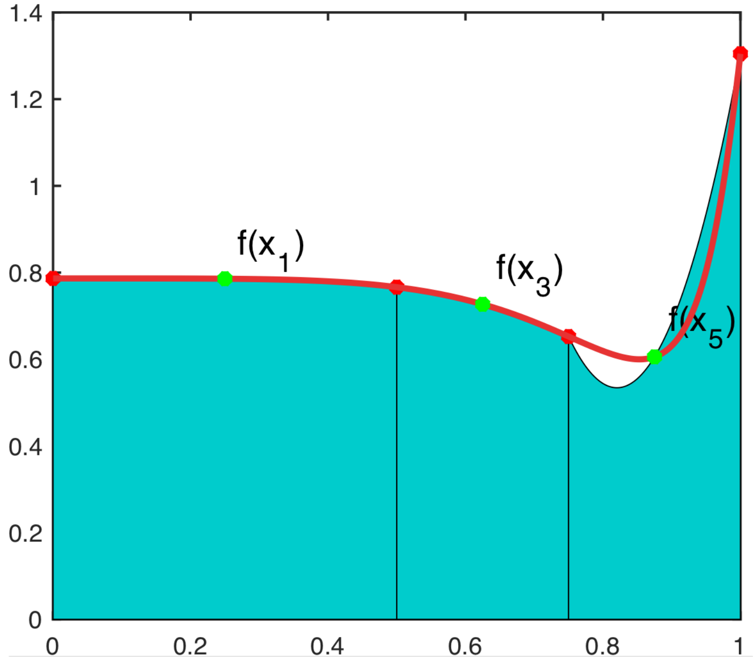
Adaptive refinement example

- The integral of the **left** hand interval here looks **good**
- The **right** interval should be **split** in more subintervals



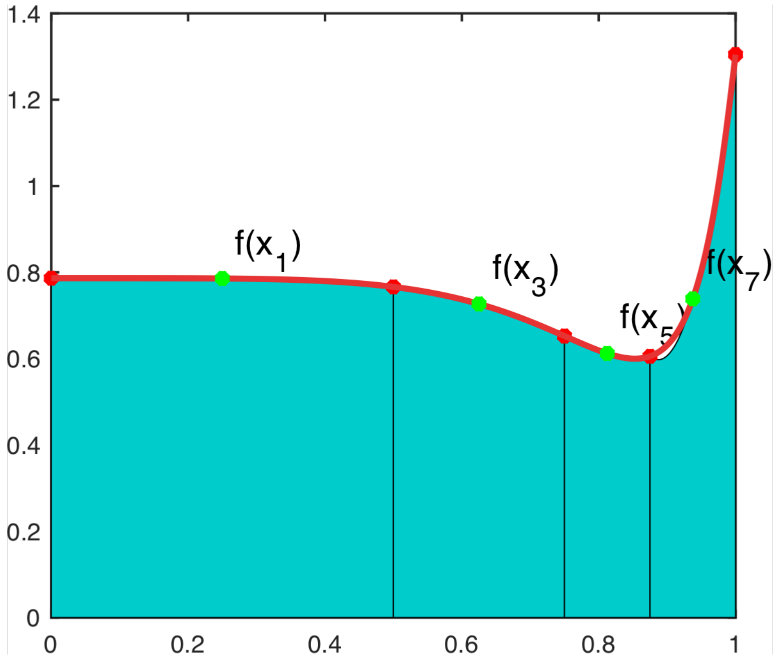
Adaptive refinement example

- The **two leftmost** integrals look **good**
- The **right** interval should be **split** in more subintervals



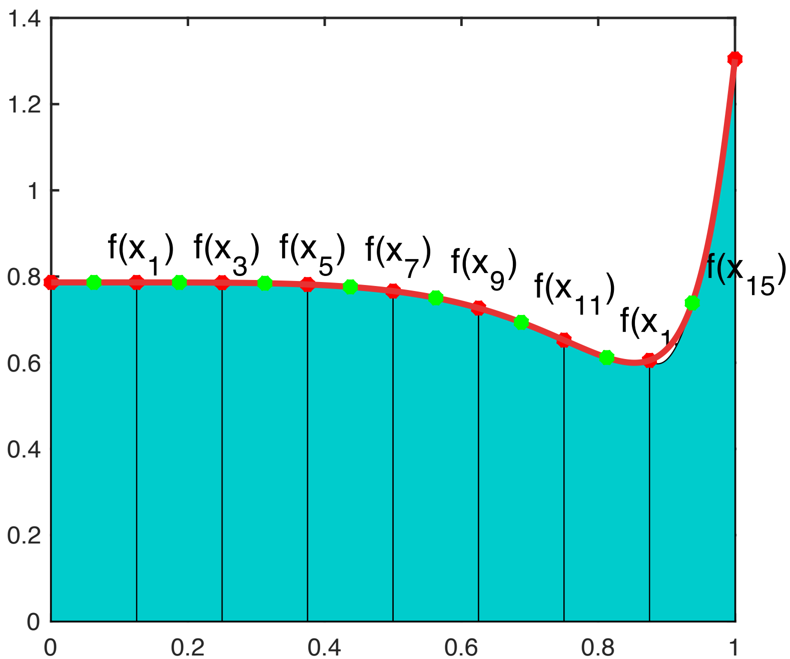
Adaptive refinement example

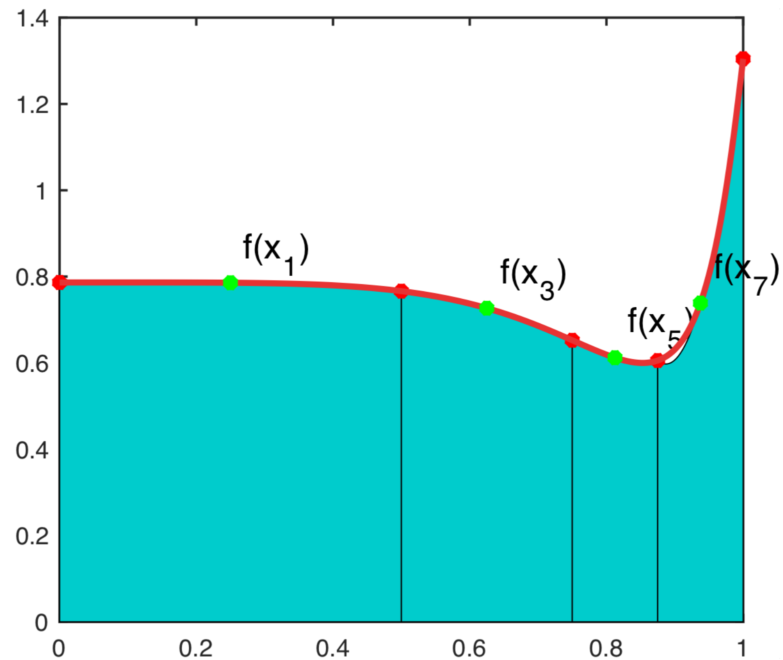
- The **whole** integral in general looks **good**
- **No** interval needs to be **split** in more subintervals



Adaptive refinement example

- *Same(ish) approximation* for **half** the number of function evaluations of the **composite** algorithm (below left)
- We must express the process as an algorithm!





Adaptive refinement

- Two clarifications are needed to state the algorithm
  - How do we keep track of which intervals are good and bad?
    - Answer: Use **recursion** and we *don't need to worry about it*
    - Make an algorithm valid for a **general interval**, then when splitting in two, simply ***apply the same algorithm to each part***
  - How do we quantify a bad interval?
    - Answer: Use **error estimates**!
      - Choose an ϵ. If the integral error over the interval is $< \epsilon$, it's good
      - If not, it's bad. Split the interval in two and require error $< \epsilon/2$ in each
    - An **added bonus**: We can **guarantee** an error of less than ϵ.
    - Problem: We don't want to calculate the maximum derivative of f!
      - With a clever trick, we can estimate it instead!

Automatic error estimates

- Write $S(a,b)$ for the (non-composite) Simpson's rule. Simpson's rule has an **error term**:
$$\int_a^b f(x)\mathrm{d}x \approx S(a,b) + M(b-a)^5.$$
  - The value of M is **unknown** here.
- With $c = (b+a)/2$,
$$\int_a^c f(x)\mathrm{d}x \approx S(a,c) + \frac{M(b-a)^5}{32}$$
$$\int_c^b f(x)\mathrm{d}x \approx S(c,b) + \frac{M(b-a)^5}{32}$$
- Adding both integrals also estimates the integral from a to b

Automatic error estimates

- We have two slightly different estimates of the integral,
$$\int_a^b f(x)\mathrm{d}x \approx S(a,b) + M(b-a)^5 \approx S(a,c) + S(c,b) + \frac{M(b-a)^5}{16}.$$
- Use this to estimate the error:
$$\left|S(a,b) - \big(S(a,c) + S(c,b)\big)\right| \approx 15 * \frac{M(b-a)^5}{16}$$
- If $\left|S(a,b) - \big(S(a,c) + S(c,b)\big)\right| \leq 15 * \epsilon$, the **error** in the estimate $S(a,c) + S(c,b)$ is **smaller** than ϵ.
  - If this is true, the interval is good
  - If this is false, we split the interval in two and want an **error less than $\epsilon/2$** in each half.

A trick for even more accuracy

- We now have two estimates

$$\int_a^b f(x)\mathrm{d}x \approx S(a,b) + M(b-a)^5$$

$$\int_a^b f(x)\mathrm{d}x \approx S(a,c) + S(c,b) + \frac{M(b-a)^5}{16}$$

- If we **subtract** 1/15 of the **first** from 16/15 of the **second**:

$$\int_a^b f(x)\mathrm{d}x = \frac{16}{15}\int_a^b f(x)\mathrm{d}x - \frac{1}{15}\int_a^b f(x)\mathrm{d}x$$

$$\approx \frac{16}{15}\big(S(a,c) + S(c,b)\big) + \frac{1}{15}M(b-a)^5 - \frac{1}{15}S(a,b) - \frac{1}{15}M(b-a)^5$$

$$= \frac{16}{15}\big(S(a,c) + S(c,b)\big) - \frac{1}{15}S(a,b)$$

- **Negate** one error with another to find a **more accurate estimate**
- Eliminating errors like this is called **Richardson extrapolation**
  - General Richardson extrapolation is not curriculum, but useful

Adaptive Simpson's rule algorithm

To approximate the integral over $[a,b]$ with error $< \epsilon$:

1. Compute $S(a,b)$.
2. Compute $S(a,c)$ and $S(c,b)$.
3. Estimate the error in $S(a,c) + S(c,b)$:

   if $\left|S(a,b) - \big(S(a,c) + S(c,b)\big)\right| < 15 * \epsilon$:

   return $\frac{16}{15}\big(S(a,c) + S(c,b)\big) - \frac{1}{15} S(a,b)$

   else:

   estimate the integrals over $[a,c]$ and $[c,b]$ with error less than $\epsilon/2$
   return the two estimates added together

Summary

- The **adaptive Simpson's rule** allows us to compute integrals efficiently using two tricks:
  - **Error analysis**
    - To identify which intervals have bad estimates
    - To **improve** current estimates using **extrapolation**
    - And to do this *without having to compute derivatives*!
  - Recursion
    - Using **adaptive Simpson's rule** recursively on each subinterval
    - Exploiting the self-similarity of each subproblem

Questions about auditorium exercise

- Regarding more detailed feedback on the exercise(s):
  - Results from automatically corrected exercises (e.g. multiple choice/drag and drop) will be posted online, identifiable by candidate numbers
  - Solution proposals are posted online for comparison
  - If you need more details, you can ask an und.ass. (or stud.ass.) during lab hours
- Re-runs of the auditorium exercises in Inspera:
  - The auditorium exercises cannot be retaken
  - Exams (ordinary/continuation) from 2017 are available here: https://www.ntnu.no/wiki/display/tdt4110/Python+eksamensoppgaver

Upcoming exam preparation

- **November 30, 09:00-13:00.**
  - Check location at **studentweb** (may not be available yet)
- Theory questions will have **multiple choice** answers
  - Both numerics and programming related
  - Similar to those in **Auditorium exercise 2**
- «Formula sheet» for the exam will be **digital only**
  - We will do our best to make it as user friendly as possible

Next two weeks

- Two lectures left
  - **Repetition** and **exam prep** on November 16 and November 23!
  - I will go through the numerics from **auditorium exercise 2** in detail
  - **Suggest other topics** you want me to cover
    - Otherwise, I'll pick them myself

Questions?