

Numerical Methods I

MATH-GA 2010.001/CSCI-GA 2420.001

Benjamin Peherstorfer
Courant Institute, NYU

Based on slides by G. Stadler and A. Donev

Today

Two weeks ago

- ▶ Function approximation
- ▶ Interpolation with polynomials
- ▶ Interpolation beyond polynomials

Today

- ▶ Numerical integration
- ▶ Newton-Cotes and Gauss quadrature

Announcements

- ▶ Homework 6 is posted and due Mon, Dec 5 before class

Numerical integration

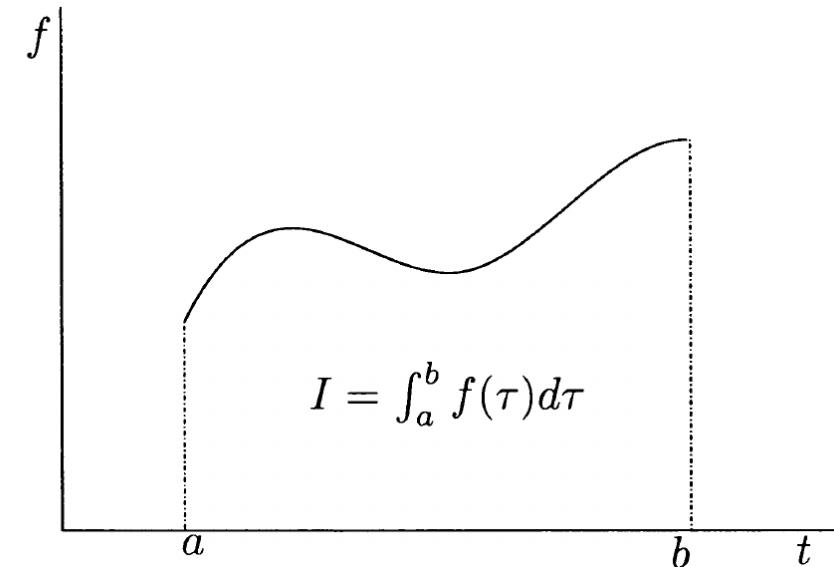
We want to approximate the definite integral

$$I(f) = I_a^b(f) = \int_a^b f(t) dt$$

numerically.

Properties of the integral:

- ▶ I is linear
- ▶ positive, i.e., if f is nonnegative, then $I(f)$ is nonnegative
- ▶ additive w.r.t. the interval bounds:
 $I_a^c = I_a^b + I_b^c$



Condition of numerical integration

Lets study the map

$$([a, b], f) \rightarrow \int_a^b f(t) dt,$$

where we use the L^1 -norm for f

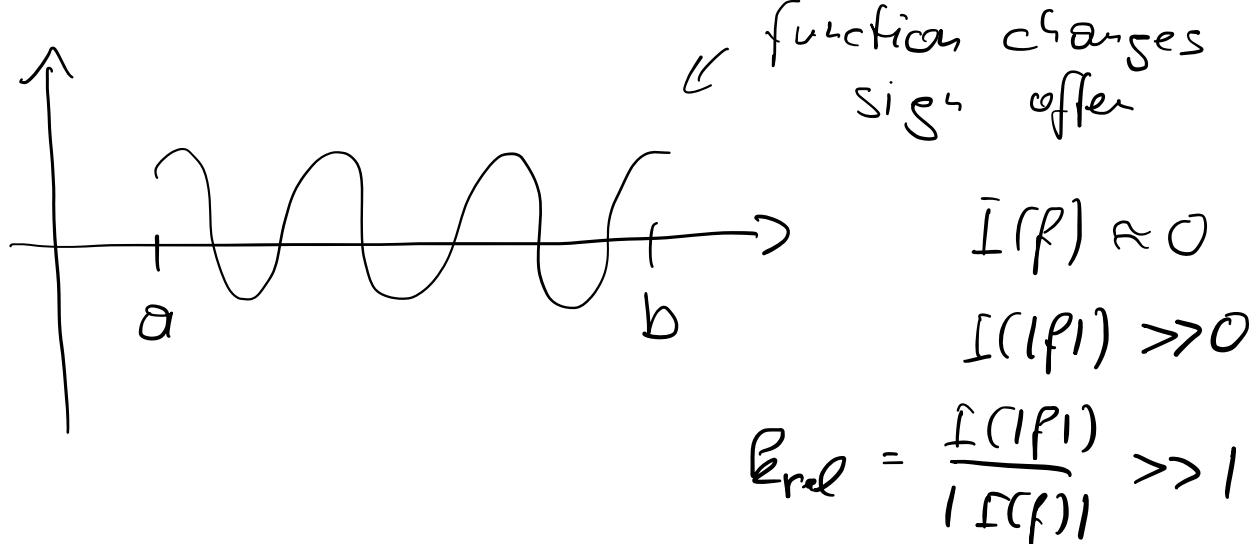
$$\|f\|_1 = \int_a^b |f(t)| dt = I(|f|)$$

The absolute and relative condition numbers of integration are:

$$\kappa_{\text{abs}} = 1,$$

$$\kappa_{\text{rel}} = \frac{I(|f|)}{|I(f)|}.$$

What does this mean?



addition

$$(a, b) \mapsto a + b$$

$$k_{\text{rel}} = \frac{|a| + |b|}{|a + b|} \quad a \approx -b \Rightarrow \text{cancellation}$$

Condition of numerical integration

Lets study the map

$$([a, b], f) \rightarrow \int_a^b f(t) dt,$$

where we use the L^1 -norm for f

$$\|f\|_1 = \int_a^b |f(t)| dt = I(|f|)$$

The absolute and relative condition numbers of integration are:

$$\kappa_{\text{abs}} = 1,$$

$$\kappa_{\text{rel}} = \frac{I(|f|)}{|I(f)|}.$$

What does this mean? Integration is harmless w.r.t. the absolute condition number, and problematic w.r.t. the relative condition number if $I(f)$ is small and f changes sign
~~~ board

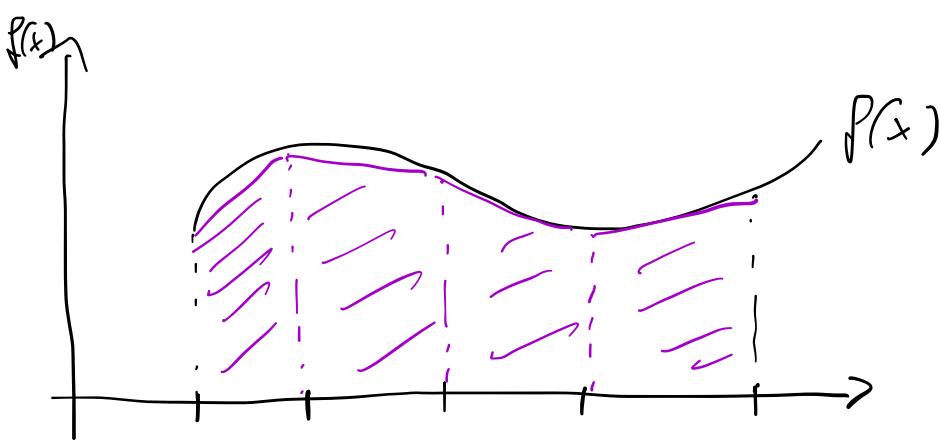
# Numerical integration

We are looking for a map

$$\hat{I} : \begin{cases} C([a, b]) & \rightarrow \mathbb{R} \\ f & \mapsto \hat{I}(f) \end{cases}$$

such that the integration error  $|I(f) - \hat{I}(f)|$  is small.

What ideas come to your mind?



$$t_0 = a \quad t_1 \quad t_2 \quad t_3 \quad \dots \quad b = t_n$$

$$\theta = t_0 < t_1 < \dots < t_n = b$$



$$T^{(n)} = \sum_{i=1}^n T_i, \quad T_i = \frac{h_i}{2} (f(t_{i-1}) + f(t_i))$$

Lower and upper sums

$$R_{\min}^{(n)} = \sum_{i=1}^n h_i \min_{t \in [t_{i-1}, t_i]} f(t), \quad R_{\max}^{(n)} = \sum_{i=1}^n h_i \max_{t \in [t_{i-1}, t_i]} f(t)$$

$$R_{\min}^{(n)} \leq T^{(n)} \leq R_{\max}^{(n)}$$



$$I(f)$$



$$I(f)$$

$$\Rightarrow T^{(n)} \xrightarrow{n \rightarrow \infty} I(f)$$

# Numerical integration

We are looking for a map

$$\hat{I} : \begin{cases} C([a, b]) & \rightarrow \mathbb{R} \\ f & \mapsto \hat{I}(f) \end{cases}$$

such that the integration error  $|I(f) - \hat{I}(f)|$  is small.

What ideas come to your mind? Example: Trapezoidal sum  $\rightsquigarrow$  board

# Numerical integration

We are looking for a map

$$\hat{I} : \begin{cases} C([a, b]) & \rightarrow \mathbb{R} \\ f & \mapsto \hat{I}(f) \end{cases}$$

such that the integration error  $|I(f) - \hat{I}(f)|$  is small.

What ideas come to your mind? Example: Trapezoidal sum  $\rightsquigarrow$  board

General quadrature formula:

$$\hat{I}(f) = \sum_{i=0}^n \lambda_i f(t_i),$$

with **weights**  $\lambda_i$  and **nodal points**  $t_i$ ,  $i = 0, 1, \dots, n$ .

## Newton-Cotes formulas

Trapezoidal rule replaces  $f$  by easy-to-integrate piecewise linear approximation  $\hat{f}$ .  
What other easy-to-integrate approximations could we use?

## Newton-Cotes formulas

Trapezoidal rule replaces  $f$  by easy-to-integrate piecewise linear approximation  $\hat{f}$ .  
What other easy-to-integrate approximations could we use?

Given **fixed** nodes  $t_0, \dots, t_n$ , use polynomial approximation

$$\hat{f} = P_f(t|t_0, \dots, t_n) = \sum_{i=0}^n f(t_i) L_{in}(t)$$

with Lagrange polynomials  $L_{0n}, \dots, L_{nn}$

Thus:

$$\hat{I}(f) = (b - a) \sum_{i=0}^n \lambda_{in} f(t_i),$$

where

$$\lambda_{in} = \frac{1}{b - a} \int_a^b L_{in}(t) dt$$

## Newton-Cotes formulas (cont'd)

Quadrature formulas defined in this way are exact for polynomials  $P \in \mathbb{P}_n$  of degree less than or equal to  $n$

$$\hat{I}(P) = I(P_n(P)) = I(P), \quad \text{for all } P \in \mathbb{P}_n$$

**Theorem:** For  $(n + 1)$  pairwise distinct nodes  $t_0, \dots, t_n$ , there exists exactly one quadrature formula (i.e., unique weights  $\lambda_0, \dots, \lambda_n$ )

$$\hat{I}(f) = (b - a) \sum_{i=0}^n \lambda_i f(t_i),$$

that is exact for all  $p \in \mathbb{P}_n$ .

Proof ↵ board

Given  $\hat{I}$  exact for  $p \in P_n$

Lagrange polynomials for  $t_0, \dots, t_n$

$$L_{in} \in P_n$$

$$\begin{aligned} I(L_{in}) &= \hat{I}(L_{in}) = (b-a) \sum_{j=0}^n z_j \underbrace{L_{in}(t_j)}_{\delta_{ij}} \\ &= (b-a) z_j \end{aligned}$$

$$z_j = \frac{1}{(b-a)} I(L_{jn})$$

$\Rightarrow$  unique weights

## Equidistantly spaced nodes

$$h_i = h = \frac{b - a}{n}, \quad t_i = a + ih, \quad i = 0, \dots, n$$

then quadrature formulas are called the *Newton-Cotes formulas* with weights

$$\lambda_{in} = \frac{1}{b-a} \int_a^b \prod_{i \neq j} \frac{t-t_i}{t_i-t_j} dt = \frac{1}{n} \int_0^n \prod_{i \neq j} \frac{s-j}{i-j} ds$$

These weights are independent of the interval boundaries  $a$  and  $b$  and can be pre-computed once and for all:

Table 9.1. Newton-Cotes weights  $\lambda_{in}$  for  $n = 1, \dots, 4$ .

| $n$ | $\lambda_{0n}, \dots, \lambda_{nn}$                                                           | Error                            | Name                                 |
|-----|-----------------------------------------------------------------------------------------------|----------------------------------|--------------------------------------|
| 1   | $\frac{1}{2} \quad \frac{1}{2}$                                                               | $\frac{h^3}{12} f''(\tau)$       | Trapezoidal rule                     |
| 2   | $\frac{1}{6} \quad \frac{4}{6} \quad \frac{1}{6}$                                             | $\frac{h^5}{90} f^{(4)}(\tau)$   | Simpson's rule, Kepler's barrel rule |
| 3   | $\frac{1}{8} \quad \frac{3}{8} \quad \frac{3}{8} \quad \frac{1}{8}$                           | $\frac{3h^5}{80} f^{(4)}(\tau)$  | Newton's 3/8-rule                    |
| 4   | $\frac{7}{90} \quad \frac{32}{90} \quad \frac{12}{90} \quad \frac{32}{90} \quad \frac{7}{90}$ | $\frac{8h^7}{945} f^{(6)}(\tau)$ | Milne's rule                         |

**Lemma** Let  $f \in C^2([a, b])$  be a twice continuously differentiable function. Then the integration error of the trapezoidal rule

$$T = \frac{b-a}{2}(f(a) + f(b))$$

with step size  $h = b - a$  can be expressed by

$$T - \int_a^b f = \frac{(b-a)^3}{12} f''(\tau),$$

for some  $\tau \in [a, b]$

Proof ↵ board

Integration errors of other Newton-Cotes formulas are listed in the table on the previous slide.

Error of trapezoidal rule:

Linear interpolant  $P \in P$ ,

$$f(t) = P(t) + [a, b, t]P(t-a)(t-b)$$

$f \in C^2$ , then

$$[a, b, t]P = \frac{P''(\tau(t))}{2}, \quad \tau(t) \in [a, b]$$

$$\int_a^b f = \int_a^b P(t) dt + \int_a^b [a, b, t]P \underbrace{(t-a)(t-b)}_{\leq 0} dt$$
$$0 \leq t \leq b$$

Because  $g(t) = (t-a)(t-b)$  has one sign over  $[a, b]$

$\exists \tilde{t} \in [a, b]$  such that ( $\rightsquigarrow$  Deaf(hard))

$$\int [a, b, t]P g(t) = [\tilde{t}, a, b]P \int g(t)$$

$$\Rightarrow \int_a^b f = T + \frac{P''(\tau(\tilde{t}))}{2} \int_a^b (t-a)(t-b) dt - \frac{(b-a)^3}{6}$$

+ trapezoidal rule

$$\Rightarrow T - \int_0^b f = \frac{(b-a)^3}{12} f''(c), \quad c \in [a,b]$$

We use equidistantly spaced nodes?

---

\*almost equal because the  $f$  does not change too much (smoothness) from  $t_i$  to  $t_{i+1}$

We use equidistantly spaced nodes? We know that this leads to poorly conditioned interpolation problems!

---

\*almost equal because the  $f$  does not change too much (smoothness) from  $t_i$  to  $t_{i+1}$

We use equidistantly spaced nodes? We know that this leads to poorly conditioned interpolation problems!

Recall: Polynomial interpolation can lead to Runge's phenomenon  $\rightsquigarrow$  high frequency oscillations  $\rightsquigarrow$  we saw that the relative condition number of numerical integration can increase for oscillatory functions

Another point of view: The weights  $\lambda_{n0}, \dots, \lambda_{nn}$  of Newton-Cotes formulas can become *negative* for larger  $n \rightsquigarrow$  cancellation because we subtract “almost equal numbers\*” in

$$\hat{I}(f) = (b - a) \sum_{i=0} \lambda_i f(t_i)$$

Weights are positive up to order 7, then some start to become negative.

---

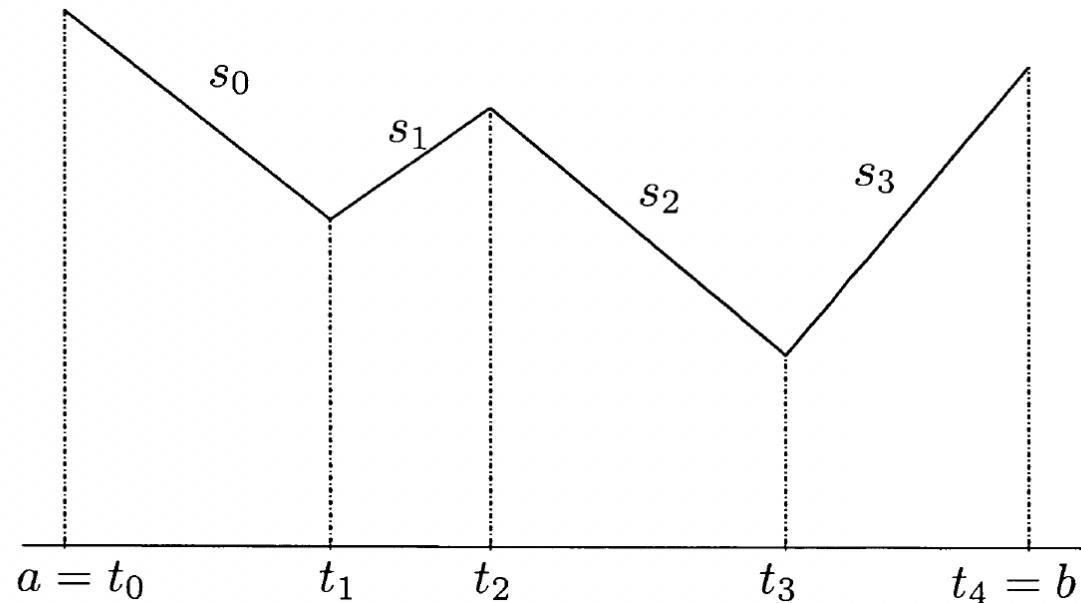
\*almost equal because the  $f$  does not change too much (smoothness) from  $t_i$  to  $t_{i+1}$

## Trapezoidal sums

To avoid poorly conditioned problems, let us split the integration interval  $[a, b]$  into  $n$  sub-intervals  $[t_{i-1}, t_i]$ ,  $i = 1, \dots, n$ . Then consider the rule

$$\hat{I}(f) = \sum_{i=1}^n \hat{I}_{t_{i-1}}^{t_i}(f),$$

where  $\hat{I}_{t_{i-1}}^{t_i}$  is a quadrature formula on the interval  $[t_{i-1}, t_i]$ .



We have seen already the trapezoidal sum with  $h = (b - a)/n$

$$T(h) = \sum_{i=1}^n T_i = h \left( \frac{1}{2}(f(a) + f(b)) + \sum_{i=1}^{n-1} f(a + ih) \right)$$

It has error

$$T(h) - \int_a^b f = \frac{(b-a)h^2}{12} f''(\tau), \quad \tau \in [a, b]$$

Proof ↵ board

Trapezoidal sum

$\exists c_i \in [t_{i-1}, t_i]$  such that

$$T_i - \int_{t_{i-1}}^{t_i} f = \frac{h^3}{12} f''(c_i)$$

$$T(h) - \int_a^b f = \sum_{i=1}^n (T_i - \int_{t_{i-1}}^{t_i} f)$$

$$= \sum_{i=1}^n \frac{h^3}{12} f''(c_i)$$

$$\left[ h = \frac{b-a}{n} \right] = \sum_{i=1}^n \frac{h^2}{12} \frac{(b-a)}{n} f''(c_i)$$
$$= \frac{h^2}{12} (b-a) \underbrace{\sum_{i=1}^n f''(c_i)}$$

To get rid of  $c_i$ :

$$\min_{t \in [a, b]} f''(t) \leq \frac{1}{n} \sum_{i=1}^n f''(c_i) \leq \max_{t \in [a, b]} f''(t)$$

$$f'' \text{ is const. } \Rightarrow \exists c \in [a, b]: \frac{1}{n} \sum_{i=1}^n f''(c_i) = f''(\tau)$$

$$\frac{1}{h} \sum_{i=1}^n f'(\xi_i) = f'(\bar{x})$$

$$T(h) - \int_0^b f = \frac{(b-a)}{12} h^2 f''(\bar{x}) \quad \text{for } \bar{x} \in [\bar{a}, b]$$

$$T(1) - \int_{t_{i-1}}^{t_i} f = \frac{(t_i - t_{i-1})^3}{12} f''(\bar{x})$$

We have seen already the trapezoidal sum with  $h = (b - a)/n$

$$T(h) = \sum_{i=1}^n T_i = h \left( \frac{1}{2}(f(a) + f(b)) + \sum_{i=1}^{n-1} f(a + ih) \right)$$

It has error

$$T(h) - \int_a^b f = \frac{(b-a)h^2}{12} f''(\tau), \quad \tau \in [a, b]$$

Proof ↵ board

↪ we can increase  $n$  (and thus decrease  $h$ ) to reduce the error without increasing the degree of the underlying polynomial

## Gauss-Christoffel quadrature

Besides piecewise approximations, what else could we do to avoid issues with high-degree polynomials?

## Gauss-Christoffel quadrature

Besides piecewise approximations, what else could we do to avoid issues with high-degree polynomials? **Change the nodes**

So far, we allowed to choose weights but the nodes were given (e.g., equidistant). Now, let's allow changing the nodes  $t_0, \dots, t_n$  too.

The best we can hope for is exact interpolation up to polynomials of degree  $2n + 1$  ( $2(n + 1)$  coefficients) (based on a non-rigorous counting argument of having  $n + 1$  DoFs because of nodes  $t_0, \dots, t_n$  and  $n + 1$  DoFs because of weights  $\lambda_0, \dots, \lambda_n$ ).

Also, for generalization, we consider quadrature of weighted integrals, with a positive weight function  $\omega(t)$ :

$$I(f) = \int_a^b \omega(t)f(t) dt$$

with weight functions  $\omega(t) = 1, \omega(t) = 1/\sqrt{1 - t^2}, \dots$

## Gauss-Christoffel quadrature (cont'd)

Theorem: For  $n \in \mathbb{N}$ , consider nodes  $\tau_{0n}, \dots, \tau_{nn}$ . For an  $n$ , define  $\hat{I}$  as

$$\hat{I}_n(f) = \sum_{i=0}^n \lambda_{in} f(\tau_{in})$$

and let it be exact for polynomials  $p \in \mathbb{P}_{2n+1}$

$$\hat{I}_n(p) = \int_a^b \omega(t)p(t)dt.$$

Then, the polynomials  $\{P_n\}$  given by

$$P_{n+1}(t) = (t - \tau_{0n}) \cdot \dots \cdot (t - \tau_{nn}) \in \mathbb{P}_{n+1}$$

are orthogonal with respect to the scalar product induced by  $\omega(t)$   $\rightsquigarrow$  proof

$\hat{I}_n$  exact for poly in  $P_{2n+1}$

For  $j < n+1$ , have  $P_j P_{n+1} \in P_{2n+1}$

$\Rightarrow \hat{I}$  exact for  $P_j P_{n+1}$

$$(P_j, P_{n+1}) = \int_{\Omega} w P_j P_{n+1} = \hat{I}_n (P_j P_{n+1})$$

$$= \sum_{i=0}^n \lambda_i w P_j(\tau_{i,n}) \underline{P_{n+1}(c_{i,n})} = 0$$

## Gauss-Christoffel quadrature (cont'd)

Therefore, the nodes  $\tau_{in}$  have to be roots of pairwise orthogonal polynomials  $\{P_j\}$  of degree  $\deg(P_j) = j$

For a given  $\omega$ , we already know that the set of orthogonal polynomials  $\{P_j\}$  is unique (if leading coefficient is 1)

We also know that the roots of these polynomials are real and have to lie in  $[a, b]$  (Deuflhard, Theorem 6.5)

For each  $\omega$  and  $n \in \mathbb{N}$ , this gives us a unique set of nodes, namely the roots of the corresponding orthogonal polynomial  $P_{n+1}$

We thus have fixed  $n + 1$  degrees of freedom so far...

However, because we want to be able to exactly integrate polynomials up to degree  $2n + 1$ , we already know that to even achieve exactness up to degree  $n$ , the weights are fixed for given nodes  $t_0, \dots, t_n$ :

$$\lambda_{in} = \frac{1}{b-a} \int_a^b L_{in}(t) dt, \quad \text{Lagrange poly } L_{in}(\tau_{jn}) = \delta_{ij}$$

(“For  $n + 1$  pairwise distinct nodes, there exists only one quadrature formula that exactly integrates polynomials up to degree  $n$ .”)

**Theorem:** Let  $\tau_{0n}, \dots, \tau_{nn}$  be the roots of the  $(n + 1)$ st orthogonal polynomial for the weight  $\omega$ . Then any quadrature formula  $\hat{I}$  is exact for polynomials up to order  $n$  if and only if it is exact up to order  $2n + 1$ .

**Proof**

$\zeta_0, \dots, \zeta_n$  roots of  $(n+1)$ -st ortho poly  $P_{n+1}$ ,  
with weight  $w$ . Then any quad rule

$$\tilde{I}_n(f) = \sum_{i=0}^n z_i f(\zeta_{i,n})$$

so satisfies

$$\tilde{I}_n \text{ exact on } P_n \Leftrightarrow \tilde{I}_n \text{ exact on } P_{n+1}.$$

$\Rightarrow$ :  $P \in P_{n+1}$ , then there exist polynomials  
 $Q, R \in P_n$  such that

$$P = Q P_{n+1} + R$$

$P_{n+1}$  is ortho. to  $P_n$  w.r.t.  $w$

$$\int_0^b w P = \underbrace{\int_0^b w Q P_{n+1}}_{=0} + \int_0^b w R = \int_0^b w R = \tilde{I}(R)$$

Additionally  $P_{n+1}(\zeta_{i,n}) = 0$

$$\tilde{I}(R) - \sum_{i=0}^n z_{i,n} R(\zeta_{i,n}) =$$

$$= \sum_{i=0}^n z_{i,n} (\underbrace{Q(\zeta_{i,n}) P_{n+1}(\zeta_{i,n})}_{=0} + R(\zeta_{i,n})) - \tilde{I}_n(P)$$

## Weight functions for Gauss-Christoffel quadrature

| $\omega(t)$              | Interval $I = [a, b]$ | Orthogonal polynomials      |
|--------------------------|-----------------------|-----------------------------|
| $\frac{1}{\sqrt{1-t^2}}$ | $[-1, 1]$             | Chebyshev polynomials $T_n$ |
| $e^{-t}$                 | $[0, \infty]$         | Laguerre polynomials $L_n$  |
| $e^{-t^2}$               | $[-\infty, \infty]$   | Hermite polynomials $H_n$   |
| 1                        | $[-1, 1]$             | Legendre polynomials $P_n$  |

Corresponding quadrature rules are usually prefixed with “Gauss-”, i.e., “Gauss-Legendre quadrature”, or “Gauss-Chebyshev quadrature”.

## Summary of Gauss quadrature

There exist uniquely determined nodes  $\tau_{0n}, \dots, \tau_{nn}$  and weights  $\lambda_{0n}, \dots, \lambda_{nn}$  such that the quadrature formula

$$\hat{I}_n(f) = \sum_{i=0}^n \lambda_{in} f(\tau_{in})$$

integrates exactly all polynomials of degree less than or equal to  $2n + 1$ , i.e.,

$$\hat{I}_n(P) = \int_a^b \omega P, \quad P \in \mathbb{P}_{2n+1}.$$

The nodes  $\tau_{in}$  are the roots of the  $n + 1$ -st orthogonal polynomial  $P_{n+1}$  with respect to the weight function  $\omega$  and the weights are

$$\lambda_{in} = \frac{1}{b-a} \int_a^b L_{in}(t) dt,$$

with the Lagrange polynomials  $L_{in}(\tau_{jn}) = \delta_{ij}$ . Furthermore, the weights are all positive  $\lambda_{in} > 0$ .

## Summary of Gauss quadrature (cont'd)

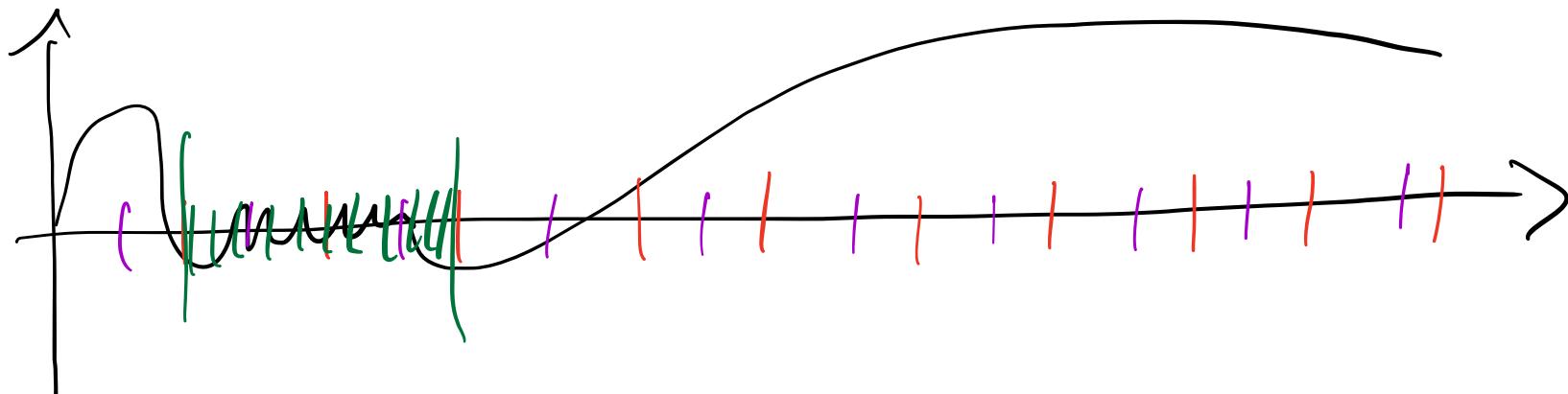
- ▶ Gauss quadrature gives the highest degree  $2n + 1$  of polynomials that can be exactly integrated with  $n + 1$  function evaluations in general
- ▶ Accuracy in Gauss-(Chebyshev, Laguerre, Hermite, Legendre,...) w.r.t. to polynomial degree can only be improved by increasing number of points; not by better weights
- ▶ Of particular interest are quadrature points for infinite intervals (Laguerre, Hermite)
- ▶ Interval partitioning superior, but only possible for  $\omega \equiv 1$  (Gauss-Legendre); otherwise weight function is different in each sub-interval
- ▶ Gauss quadrature is typically used in finite-element approximation to integrate over a local element. However, in many other cases in scientific computing, interval partitioning is superior.

# Adaptive interval partitioning

Idea: On each sub-interval, estimate the quadrature error by either:

- ▶ Using a higher-order quadrature (e.g., Simpson rule), or
- ▶ Comparing the error on a subinterval with the error on a refinement

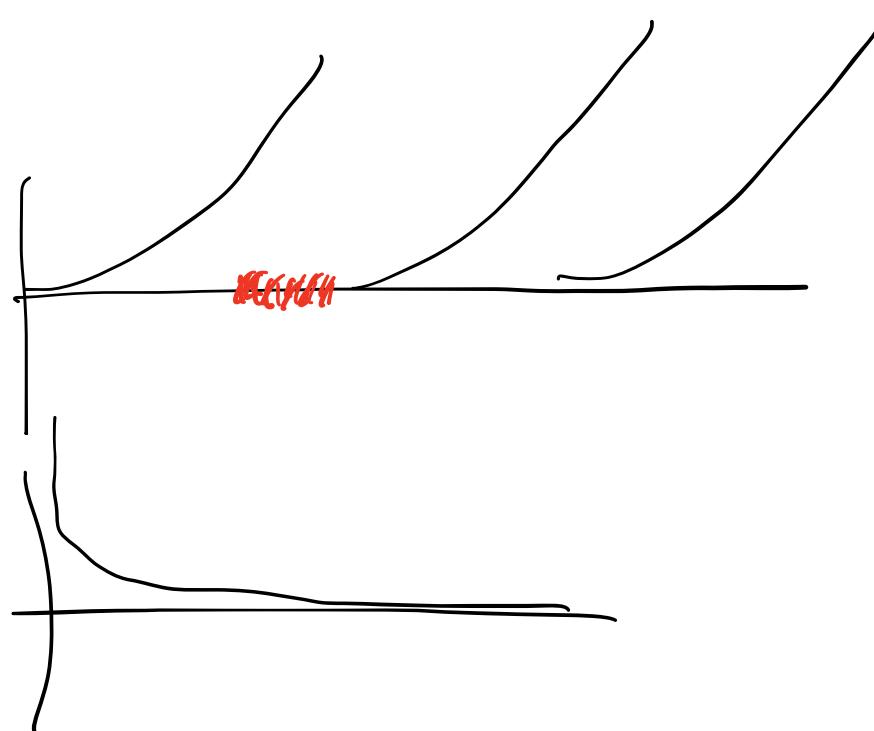
Then, subdivide the interval depending on the error estimation, and repeat  $\rightsquigarrow$  visualization!



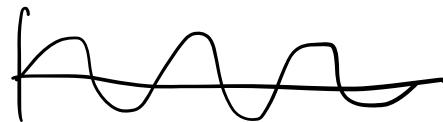
Such a method refines the nodes **a posteriori** (after having seen the function to integrate).

## Difficult cases for numerical integration

- ▶ (Unknown) discontinuities in  $f$ : adaptive quadrature continues to refine, which can be used to localize discontinuities



- ▶ Highly oscillating integrals



- ▶ (Weakly) singular integrals

## Integration in higher dimensions

A separable integral can be integrated dimension-wise:

$$J = \int_a^b \int_a^b \phi(x, y) dx dy = \int_a^b \phi^{(x)}(x) dx \int_a^b \phi^{(y)}(y) dy,$$

where

$$\phi(x, y) = \phi^{(x)}(x) \phi^{(y)}(y) \quad (1)$$

Recall that one idea of numerical quadrature is to replace  $f$  with an easy-to-integrate  $\hat{f}$

- ▶ Choose a basis  $\phi_1(x, y), \dots, \phi_n(x, y)$  and approximate

$$f(x, y) \approx \sum_{i=1}^n c_i \phi_i(x, y),$$

and choose  $\phi_i(x, y)$  such that (1) holds.

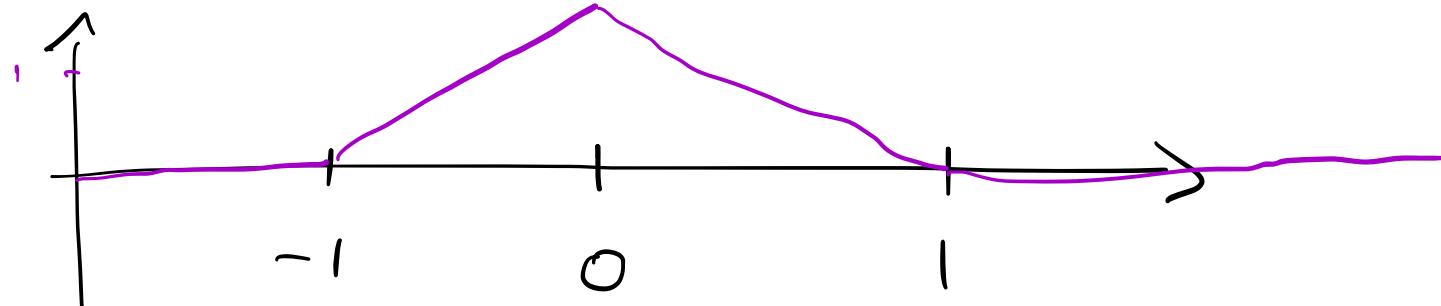
- ▶ Then integrate

$$I(f) \approx \sum_{i=1}^n c_i \hat{I}(\phi_i^{(x)}) \hat{I}(\phi_i^{(y)})$$

One way to build such a basis is via tensor products of linear functions  
(multi-dimensional analog to piecewise linear quadrature)

Define the “mother hat” function

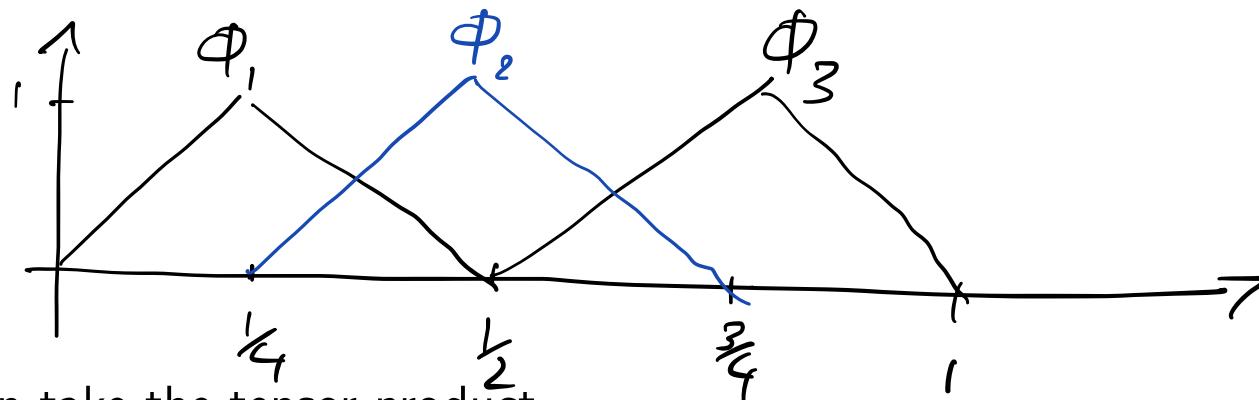
$$\underline{\phi(x) = \max(1 - |x|, 0)}, \quad x \in \mathbb{R}$$



If we want  $n = \underline{2^{\ell} + 1}$  basis functions, then translate and dilate it to center it on the grid points

$$\underline{x_i} = \underline{i2^{-\ell}} \quad i = 0, \dots, n$$

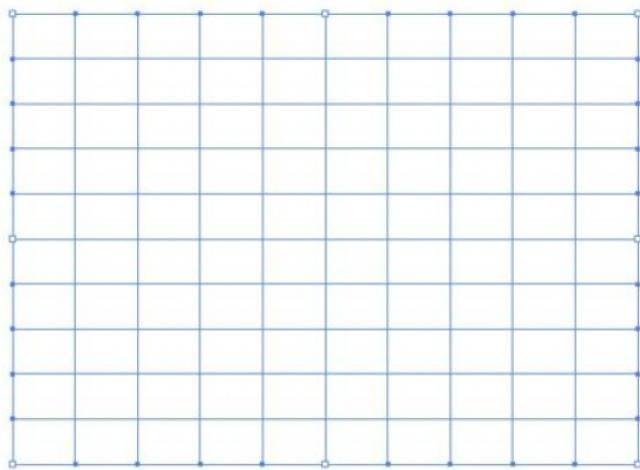
$$\underline{\phi_i(x)} = \underline{\phi}\left(\frac{x - \underline{x_i}}{2^{-\ell}}\right)$$



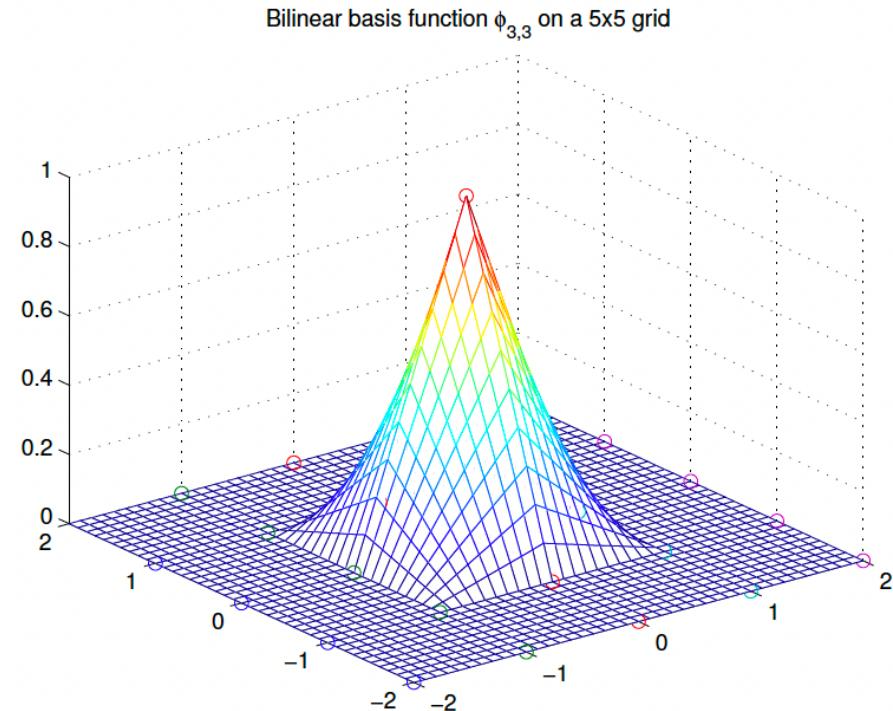
Then take the tensor product

$$\phi_{i,j}(x, y) = \phi_i(x)\phi_j(y), \quad i, j = 0, \dots, n$$

lead to a basis of the piecewise bilinear functions in two dimensions



[Figure: A. Donev]



The basis  $\{\phi_{ij}\}$  is a nodal point basis for piecewise bilinear functions in  $[0, 1]^2 \rightsquigarrow$

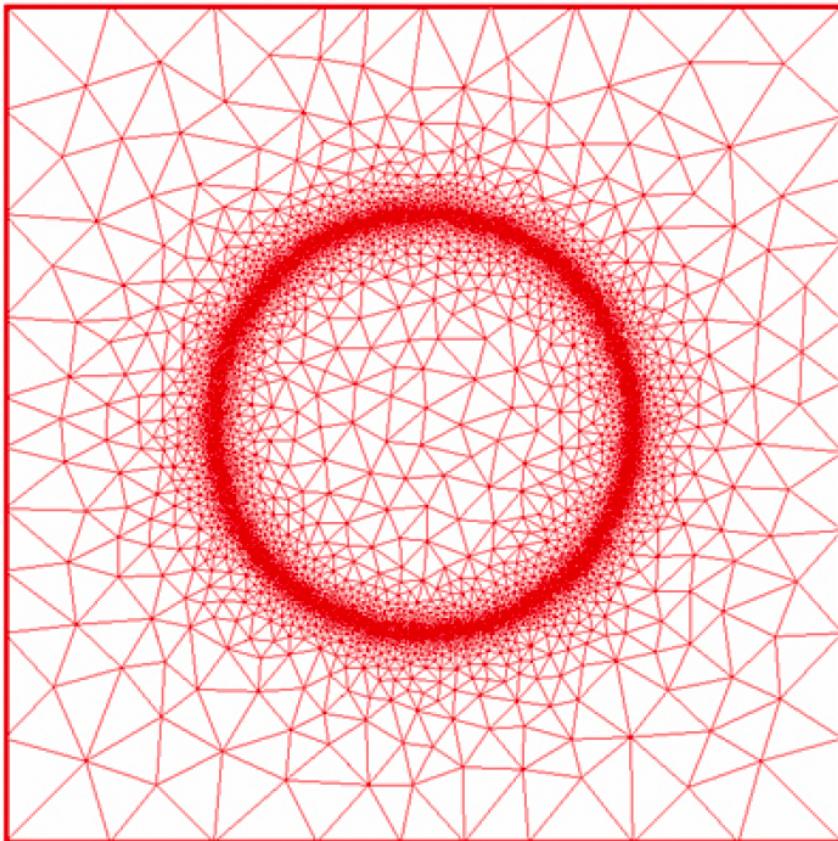
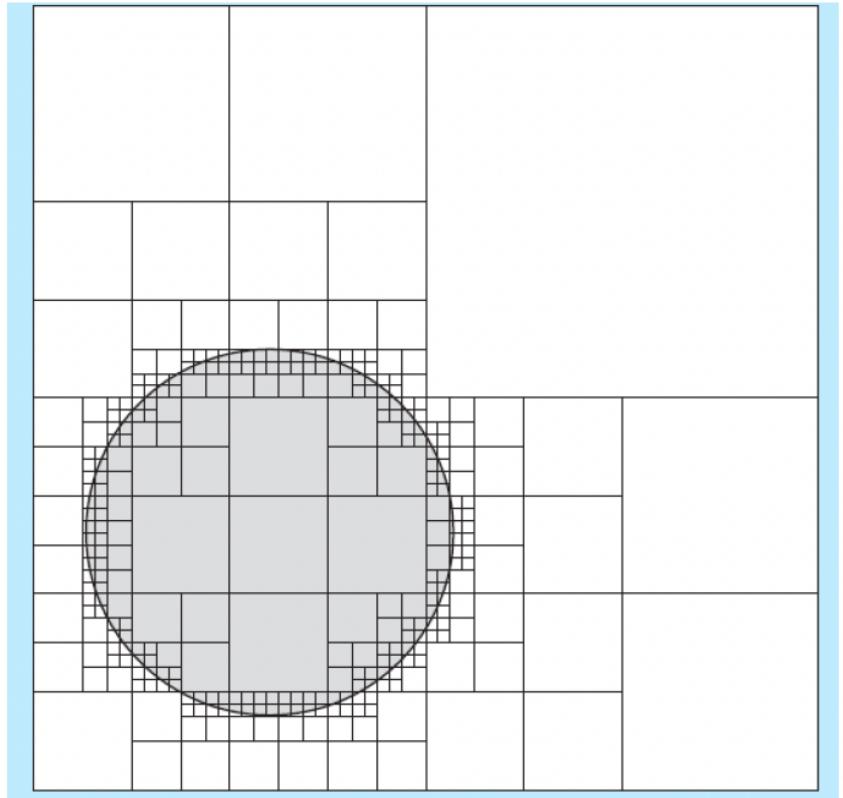
$$\hat{f}(x, y) = \sum_{i,j} \underline{f(x_i, y_j)} \phi_{ij}(x, y)$$

interpolates  $f$  at the nodes  $\{(x_i, y_j)\}_{i,j=0}^n$

Integrate the bilinear basis functions  $\{\phi_{ij}\}$  to obtain the weights  $\lambda_{ij}$  and then approximate

$$\hat{I}(f) = \sum_{i,j} f(x_i, y_i) \lambda_{ij},$$

In multiple dimensions, adaptivity is essential to keep the computational costs manageable



[Figure: A. Donev]

## Conclusions

- ▶ Numerical integration approximates an integral via a discrete weighted sum of function values over a set of nodes
- ▶ Numerical integration typically is based on function approximation
- ▶ Numerical integration with Newton-Cotes only for low degrees (trapezoidal, Simpson). Gaussian integration based on special set of nodes and weights (orthogonal polynomials) that also works for higher orders
- ▶ For higher accuracy, split the interval in sub-intervals and compute piecewise polynomial approximations (trapezoidal sum, Simpson sum). Can reduce the error without requiring smoother functions
- ▶ Integration in high dimensions becomes exponentially harder because the number of grid points grows as  $N^d \rightsquigarrow$  curse of dimensionality

$$\int_0^1 4x(1-x)dx = \frac{8}{3}$$

