

Algorithms for Scientific Computing

Hierarchical Methods and Sparse Grids
– 1D Hierarchical Basis –

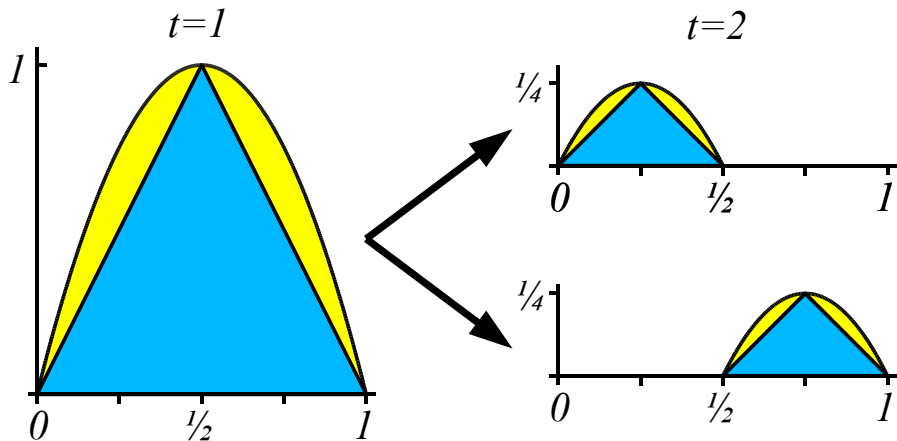
Michael Bader
Technical University of Munich
Summer 2018



TUM Uhrenturm

Archimedes' Quadrature

Compute an approximation of $F_1 := \int_0^1 4 \cdot x \cdot (1 - x) dx = \frac{2}{3}$



Archimedes' Quadrature (2)

- Integrating $4x(1-x)$, we have to consider several quantities
- Ordered by (recursive) level t :

Level-depth	1	2	3	4	...	t
Mesh-width h	$1/2$	$1/4$	$1/8$	$1/16$...	2^{-t}
# triangles	1	2	4	8	...	$\frac{1}{2}2^t$
surplus v	1	$1/4$	$1/16$	$1/64$...	$4 \cdot 2^{-2t}$
Area of triangle D_1	$1/2$	$1/16$	$1/128$	$1/1024$...	$4 \cdot 2^{-3t}$
Sum (current t)	$1/2$	$1/8$	$1/32$	$1/128$...	$2 \cdot 2^{-2t}$
Sum ($\leq t$)	$1/2$	$5/8$	$21/32$	$85/128$...	$\frac{2}{3}(1 - 2^{-2t})$
Error	$1/6$	$1/24$	$1/96$	$1/384$...	$\frac{2}{3}2^{-2t}$

Approximation of Functions

- Goal: analyze Archimedes' quadrature rule for more general functions
- We need a representation of the (approximating) function $u(x)$:
 - u as linear combination of ansatz functions ϕ_i :

$$u(x) = \sum_{i=1}^N \alpha_i \cdot \phi_i(x)$$

- Integrating $u(x)$:

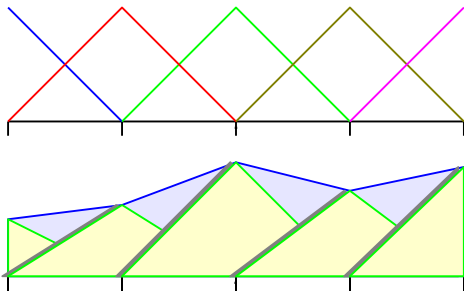
$$\int_a^b u(x) dx = \sum_{i=1}^N \alpha_i \int_a^b \phi_i(x) dx,$$

- Weighted sum of α_i
- Compare to Newton-Cotes: integrate the interpolant (polynomial)
 - ⇒ leads to a weighted sum of function evaluations

Composite Trapezoidal Rule: Function

Interpolant

- Continuous, piecewise linear function
- Represent u in nodal point (hat) basis



- Coefficients α_i are function values at grid points
- Basis functions have area h ($h/2$ at boundaries)

Piecewise Linear Functions

Ansatz space and basis functions

- Only consider $u : [0, 1] \rightarrow \mathbb{R}$
- Consider discretization level $n \in \mathbb{N}$
- Mesh-width $h_n = 2^{-n}$
- Grid points $x_{n,i} = i \cdot h_n$
- Define “mother of all hat functions”

$$\phi(x) := \max\{1 - |x|, 0\}$$

⇒ Basis functions

$$\phi_{n,i}(x) := \phi\left(\frac{x - x_{n,i}}{h_n}\right)$$

- Nodal point basis $\Phi_n := \{\phi_{n,i}, 0 \leq i \leq 2^n\}$

Piecewise Linear Functions (2)

Towards Function Spaces:

- Space of continuous piecewise linear functions:

$$V_n = \text{span}(\Phi_n)$$

- Interpolants $u_n \in V_n$:

$$u_n(x) = \sum_{i=0}^{2^n} \alpha_{n,i} \phi_{n,i}(x)$$

- V_n the space of all such interpolants u_n

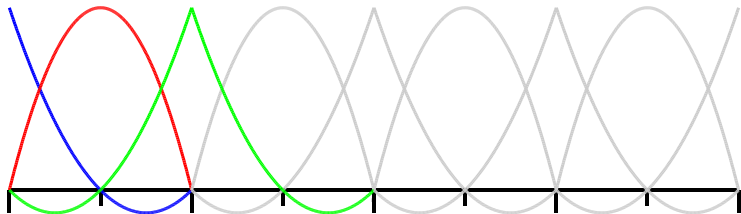
Interpolation with Nodal Basis:

- Interpolation conditions: $u_n(x_j) = \sum \alpha_{n,i} \phi_{n,i}(x_j) \stackrel{!}{=} f(x_{n,j})$
- Due to nodal basis: $\phi_{n,i}(x_j) = 0$ if $i \neq j$, and $\phi_{n,j}(x_j) = 1$
- Thus: all $\alpha_{n,j} = f(x_{n,j})$

Composite Simpson's Rule: Function

Interpolant

- Continuous, piecewise quadratic function
- More complicated basis:



- Basis functions: Lagrangian polynomials, glued together
- α_j : function values at grid points
- Basis functions have area $h/6$ (blue), $4h/6$ (red), $2h/6$ (green)
- We'll not formally define basis functions here ...

From Composite Trapezoidal to Archimedes

Piecewise linear functions

- We restrict our functions u to $u(0) = u(1) = 0$
- Nodal point basis for discretization level n :

$$\Phi_n := \{\phi_{n,i}, 1 \leq i \leq 2^n - 1\}$$

- Wanted: *function space*

$$V := \bigcup_{l=1}^{\infty} V_l$$

contains all functions which are in V_l for sufficiently large l

- However: generating system of V as

$$\Phi := \bigcup_{l=1}^{\infty} \Phi_l$$

does not lead to a basis (not linear independent)

Hierarchical Basis

- We are interested in a hierarchical decomposition of V_l
 \Rightarrow Define **hierarchical increment** W_l , such that V_l is a *direct sum*:

$$V_l = V_{l-1} \oplus W_l$$

Side-note: direct sum

- \rightarrow Every $u_l \in V_l$ can be uniquely decomposed as $u_l = u_{l-1} + w_l$, with $u_{l-1} \in V_{l-1}$ and $w_l \in W_l$
- W_l has to contain 2^{l-1} ansatz functions:

$$\dim V_l = 2^l - 1 = \dim V_{l-1} + \dim W_l$$

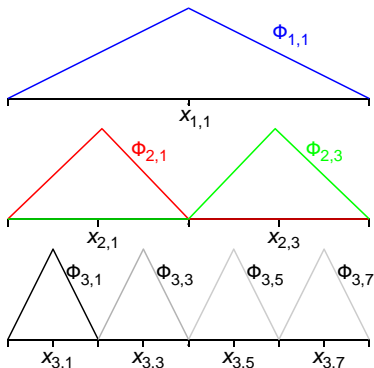
- This holds (introducing index sets \mathcal{I}_l) for

$$\mathcal{I}_l := \{i : 1 \leq i < 2^l, i \text{ odd}\}$$

$$W_l := \text{span} \{\phi_{l,i} : i \in \mathcal{I}_l\}$$

Hierarchical Increments

- Set of hierarchical increments W_l
- For $l = 1$: $W_1 = V_1$
- Example for $l = 1, 2, 3$:



Hierarchical Basis (cont.)

- Then

$$V_n = \bigoplus_{l=1}^n W_l$$

is a direct sum, too:

- $u \in V_n$ can be decomposed uniquely into $w_l \in W_l$:

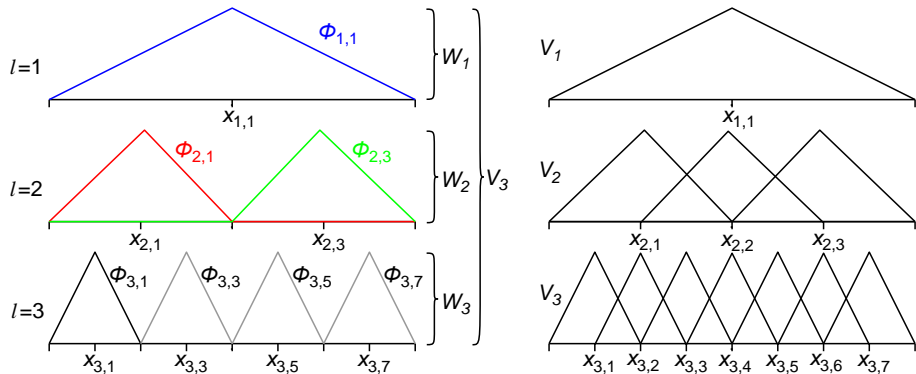
$$u = \sum_{l=1}^n w_l = \sum_{l=1}^n \sum_{i \in \mathcal{I}_l} v_{l,i} \phi_{l,i}$$

→ Coefficients $v_{l,i}$ are **hierarchical surplusses**

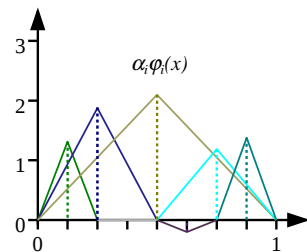
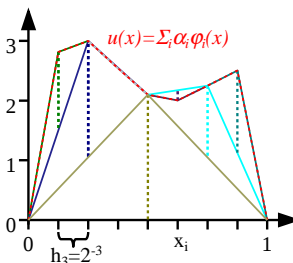
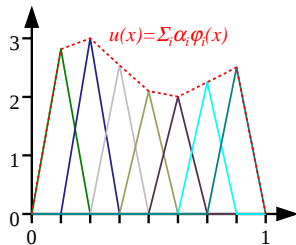
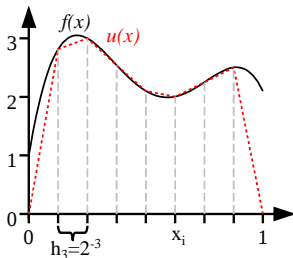
- Corresponding basis of V_n (or, with ∞ instead of n , of V)

$$\Psi_n := \bigcup_{l=1}^n \{\phi_{l,i} : i \in \mathcal{I}_l\}.$$

Comparison



Comparison (2)



Numerical Integration with Hierarchical Basis

Key Ingredients:

- Integration of $u(x)$:

$$\int_a^b u(x) dx = \int_a^b \sum_i^N \alpha_i \phi_i(x) dx = \sum_i^N \alpha_i \int_a^b \phi_i(x) dx,$$

- Using a hierarchical basis:

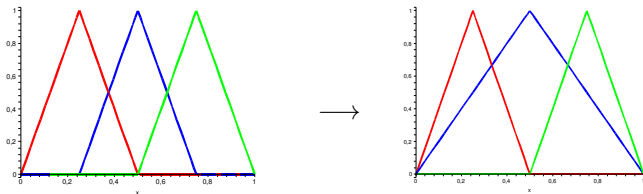
$$\int_a^b u dx = \int_a^b \sum_{l=1}^n \sum_{i \in \mathcal{I}_l} v_{l,i} \phi_{l,i} dx = \sum_{l=1}^n \sum_{i \in \mathcal{I}_l} v_{l,i} \int_a^b \phi_{l,i} dx = \sum_{l=1}^n \sum_{i \in \mathcal{I}_l} v_{l,i} h_l$$

- Computation of hierarchical surpluses:

$$v_{l,i} = u(x_{l,i}) - \frac{1}{2} (u(x_{l,i-1}) + u(x_{l,i+1}))$$

i.e., difference between function and linear interpolant (on coarser level) at $x_{l,i} \rightarrow$ **hierarchical surplus**

Hierarchical Basis Transformation



- represent “wider” hat function $\phi_{1,1}(x)$ via basis functions $\phi_{2,j}(x)$

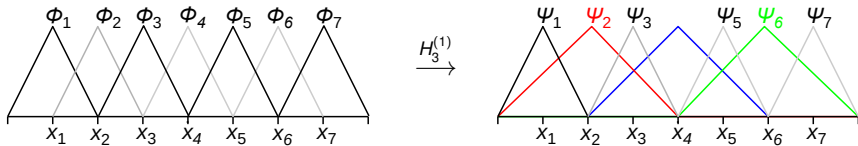
$$\phi_{1,1}(x) = \frac{1}{2}\phi_{2,1}(x) + \phi_{2,2}(x) + \frac{1}{2}\phi_{2,3}(x)$$

- consider vector of hierarchical/nodal basis functions and write transformation as matrix-vector product:

$$\begin{pmatrix} \psi_{2,1}(x) \\ \psi_{2,2}(x) \\ \psi_{2,3}(x) \end{pmatrix} := \begin{pmatrix} \phi_{2,1}(x) \\ \phi_{1,1}(x) \\ \phi_{2,3}(x) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_{2,1}(x) \\ \phi_{2,2}(x) \\ \phi_{2,3}(x) \end{pmatrix}$$

Hierarchical Basis Transformation (2)

Consider “semi-hierarchical” transform: (step 1)



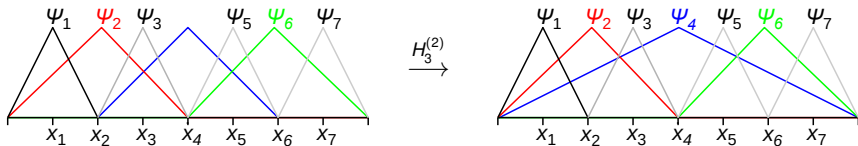
Matrices for change of basis are then: $(H_3^{(2)})$ to transform to hierarchical basis)

$$H_3^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_3^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Hierarchical Basis Transformation (2)

Consider “semi-hierarchical” transform: (step 2)



Matrices for change of basis are then: $(H_3^{(2)})$ to transform to hierarchical basis)

$$H_3^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_3^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Hierarchical Basis Transformation (3)

Level-wise hierarchical transform:

- hierarchical basis transformation: $\psi_{n,i}(x) = \sum_j H_{i,j} \phi_{n,j}(x)$
- written as matrix-vector product: $\vec{\psi}_n = H_n \vec{\phi}_n$
- $H_n \vec{\phi}_n$ can be performed as a sequence of level-wise transforms:

For k from 1 to n-1

$$\vec{\phi}_n := H_n^{(k)} \vec{\phi}_n$$

- matrix H_n for hierarchical basis transformation is thus:

$$H_n = H_n^{(n-1)} H_n^{(n-2)} \dots H_n^{(2)} H_n^{(1)}$$

- where each level-wise transform $H_n^{(k)} \vec{\phi}_n$ has a simple loop implementation:

For j from 2^k to 2^n step 2^k

$$\phi_{n,j} := \frac{1}{2} \phi_{n,j-2^{k-1}} + \phi_{n,j} + \frac{1}{2} \phi_{n,j+2^{k-1}}$$

Hierarchical Coordinate Transformation

- consider function $f(x) \approx \sum_i a_i \psi_{n,i}(x)$ represented via hier. basis
- wanted: corresponding representation in nodal basis

$$\sum_j b_j \phi_{n,j}(x) = \sum_i a_i \psi_{n,i}(x) \approx f(x)$$

- with $\psi_{n,i}(x) = \sum_j H_{i,j} \phi_{n,j}(x)$ we obtain

$$\sum_j b_j \phi_{n,j}(x) = \sum_i a_i \sum_j H_{i,j} \phi_{n,j}(x) = \sum_j \sum_i a_i H_{i,j} \phi_{n,j}(x)$$

- compare coordinates and get

$$b_j = \sum_i H_{i,j} a_i = \sum_i (H^T)_{j,i} a_i$$

- written in vector notation: $b = H^T a$

Hierarchical Coordinate Transformation (2)

- transform $b = H^T a$ turns “hierachical” coefficients a into “nodal” coefficients b :

$$\sum_j b_j \phi_{n,j}(x) = \sum_i a_i \psi_{n,i}(x) \approx f(x)$$

- Recall that $H_n = H_n^{(n-1)} H_n^{(n-2)} \dots H_n^{(2)} H_n^{(1)}$ has a level-wise representation, therefore:

$$H_n^T = \left(H_n^{(1)}\right)^T \left(H_n^{(2)}\right)^T \dots \left(H_n^{(n-2)}\right)^T \left(H_n^{(n-1)}\right)^T$$

- use loop-based implementation for $\left(H_n^{(k)}\right)^T a$ to get **fast algorithm**:

For k from n-1 downto 1

For i from 2^{k-1} to 2^n step 2^k

$$a_i := \frac{1}{2} a_{i-2^{k-1}} + a_i + \frac{1}{2} a_{i+2^{k-1}} \quad (\text{with } a_0 = a_{2^n} = 0)$$

Hierarchical Coordinate Transformation (3)

Now: transform “nodal” coefficients b into “hierachical” coefficients a

- thus: solve $H^T a = b$ for a (for given b), or $b = (H^T)^{-1} a = H^{-T} a$
- again via level-wise representation:

$$H_n^{-T} = \left(H_n^{(n-1)}\right)^{-T} \left(H_n^{(n-2)}\right)^{-T} \dots \left(H_n^{(2)}\right)^{-T} \left(H_n^{(1)}\right)^{-T}$$

- iterate over $b^{\text{new}} = H^{-T} b^{\text{old}}$ (starting with $b^{\text{old}} = a$)
or repeatedly solve $H^T b^{\text{new}} = b^{\text{old}}$
- consider example $(H_3^{(1)})^T b^{\text{new}} = b^{\text{old}}$:

$$\begin{pmatrix} 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} b_1^{\text{new}} \\ \vdots \\ b_7^{\text{new}} \end{pmatrix} = \begin{pmatrix} b_1^{\text{old}} \\ \vdots \\ b_7^{\text{old}} \end{pmatrix}$$

- for row 3 (1, 5 and 7 similar):

$$\frac{1}{2} b_2^{\text{new}} + b_3^{\text{new}} + \frac{1}{2} b_4^{\text{new}} = b_3^{\text{old}} \Leftrightarrow b_3^{\text{new}} = b_3^{\text{old}} - \frac{1}{2} b_2^{\text{new}} - \frac{1}{2} b_4^{\text{new}} = b_3^{\text{old}} - \frac{1}{2} (b_2^{\text{old}} + b_4^{\text{old}})$$

↪ **computation of hierarchical surplus!**