Analysis of an Algorithm Using the Hoare Logic

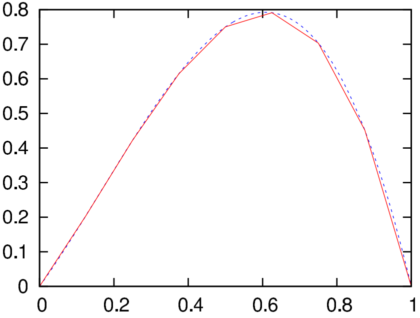
Florian Klöck

Technische Universität München

March 2007

How can we approximate functions?

We want to approximate functions $f : [a, b] \to \mathbb{R}$.
Simplification: $f : [0, 1] \to \mathbb{R}, f(0) = f(1) = 0$



"linear splines" with equidistant nodes ("lattice points") with distance $h_n = 2^{-n}, n \in \mathbb{N}$ ("mesh size")

Outline

1 Problem
- Outline of the presentation
- Bases for the space of linear splines
- Motivation

2 Conversion into the nodal point basis
- Algorithm for conversion into the nodal point basis
- Proof of the algorithm

3 Conversion into the hierarchical basis
- Algorithm for conversion into the hierarchical basis
- Proof of the algorithm

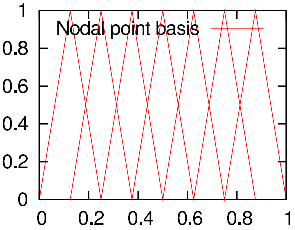
The nodal point basis

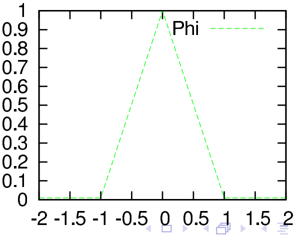
We want to find a basis for the space of linear splines $s : [0,1] \to \mathbb{R}$ with $s(0) = s(1) = 0$ and mesh size $h_n = 2^{-n}$.
The lattice points are:

$$x_{n,i} = ih_n \text{ with } i \in \{1, 2, ..., 2^n - 1\}$$

A simple basis is:

$$\bigcup_{i=1}^{2^n-1} \{\Phi_{n,i}\} \text{ with } \Phi_{n,i} := \Phi(\frac{x - x_{n,i}}{h_n}), \Phi(x) := \max\{1 - |x|, 0\}$$

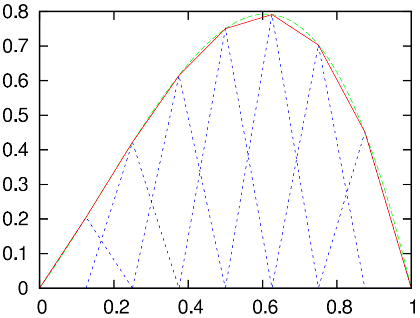




Representation in the nodal point basis

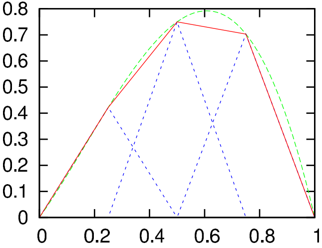
The $\Phi_{n,i}$ are piecewise linear and continuous. $\Phi_{n,i}(x_j) = \delta_{ij}$.
Piecewise linear and continuous functions are equal when they are equal on every lattice point, so s can be expressed as follows:

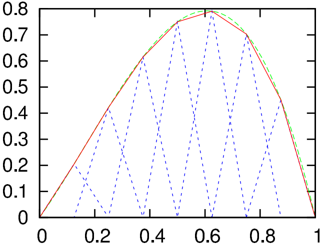
$$s(x) = \sum_{i=1}^{2^n-1} f(x_i)\Phi_{n,i}(x)$$



Quadrature

Let's assume we would use the linear spline for quadrature (=numerical integration).
If we increase n, we have to compute everything again with the nodal point basis:





We will see, that there exists a better basis for this purpose.

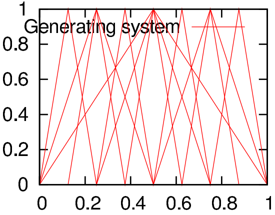
Generating system which includes the nodal point basis

Instead of a basis for the linear splines we could also use a more general generating system:

$$\bigcup_{l=1}^{n} \bigcup_{i=1}^{2^l-1} \{\Phi_{l,i}\}$$

A linear spline is represented in this generating system as follows:

$$s(x) = \sum_{l=1}^{n} \sum_{i=1}^{2^l-1} v_{l,i}\Phi_{l,i}(x) \text{ with a coefficient vector } v$$

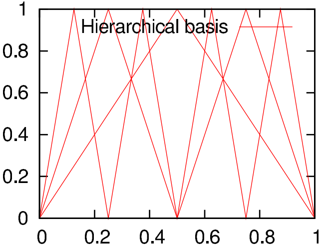


The hierarchical basis

$$\text{Generating system: } \bigcup_{l=1}^{n} \bigcup_{i=1}^{2^l-1} \{\Phi_{l,i}\}$$

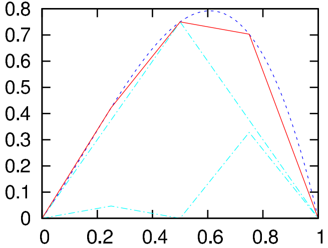
We consider the following both bases:

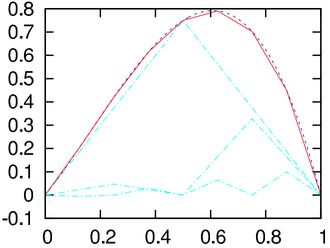
- Nodal point basis: Only $\Phi_{n,i}$ $(\bigcup_{i=1}^{2^n-1}\{\Phi_{n,i}\})$
- Hierarchical basis: Only $\Phi_{l,i}$ with odd i $(\bigcup_{l=1}^{n} \bigcup_{i\in\{1,3,5,\ldots,2^l-1\}}\{\Phi_{l,i}\})$



Quadrature with the hierarchical basis

Using the hierarchical basis, we don't need to recalculate the complete integral if we increase n:





Motivation

In dimension 1, this problem is quite harmless, but it is a prototype for very complicated problems in higher dimensions. Keywords:

- Sparse Grids
- Finite Elements

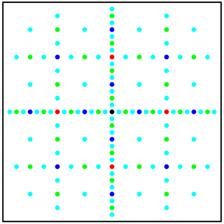


Figure: sparse grid

Representation in the nodal point basis

How can we represent the function given by the vector v in the nodal point basis? Algorithm `toNodalPointBasis`:

- input
  - integer $n > 1$
  - vector v with $\sum_{l=1}^{n} \sum_{i=1}^{2^l-1} v_{l,i} \Phi_{l,i} = u$
- output
  - vector v with $\sum_{l=1}^{n} \sum_{i=1}^{2^l-1} v_{l,i} \Phi_{l,i} = u$ and $v_{l,i} = 0$ for all $l < n$

Conversion into the nodal point basis

Algorithm for conversion of a vector v into the nodal point basis:

toNodalPointBasis

$$
\begin{array}{lll}
\textit{for} & l = 1, \dots, n-1: & \\
 & \textit{for} & i = 1, \dots, 2^l - 1: \\
 & & v_{l+1,2i-1} \mathrel{+}= v_{l,i}/2 \\
 & & v_{l+1,2i} \mathrel{+}= v_{l,i} \\
 & & v_{l+1,2i+1} \mathrel{+}= v_{l,i}/2 \\
 & & v_{l,i} = 0
\end{array}
$$

In the following we prove the correctness of the algorithm.

Revision of the Hoare rules

0. $\{A\}$ % noOperation $\{A\}$
1. Axiom of assignment: $\{A[E/x]\}x := E\ \{A\}$
2. Rule of consequence: $\dfrac{A' \Rightarrow A \text{ and } \{A\}S\{B\} \text{ and } B \Rightarrow B'}{\{A'\}S\{B'\}}$
3. Rule of composition: $\dfrac{\{A\}S\{B\} \text{ and } \{B\}T\{C\}}{\{A\}S;T\{C\}}$
4. Rule of iteration: $\dfrac{\{A \text{ and } b\}S\{A\}}{\{A\}\mathit{while}\ b\ \mathit{do}\ S\{A \text{ and } \mathit{not}(b)\}}$
5. Conditional rule: $\dfrac{\{A \text{ and } c\}S\{B\} \text{ and } \{A \text{ and } \mathit{not}(c)\}T\{B\}}{\{A\}\mathit{if}\ c\ \mathit{then}\ S\ \mathit{else}\ T\{B\}}$

Definitions

We need the following definitions for the proof:

- $f_v := \sum_{l=1}^{n} \sum_{i=1}^{2^l-1} v_{l,i} \Phi_{l,i}$
- $P_u(f) :\Leftrightarrow f \equiv u$ (u is the input function)
- $v_{l,i} := 0$ for $l > n$ or $l < 1$ or $i < 1$ or $i \geq 2^l$

Transformation of the program

The algorithm needs to be transformed so that it uses only control structures covered by the Hoare logic:

toNodalPointBasis

$l = 1$
while $l \neq n$:
  $i = 1$
  while $i \neq 2^n$:
    $v_{l+1,2i-1} = v_{l+1,2i-1} + v_{l,i}/2$
    $v_{l+1,2i} = v_{l+1,2i} + v_{l,i}$
    $v_{l+1,2i+1} = v_{l+1,2i+1} + v_{l,i}/2$
    $v_{l,i} = 0$
    $i = i + 1$
  $l = l + 1$

Proof: The outer loop

$$
\begin{array}{l}
\{P_u(f_v)\} \\
l = 1 \\
\{P_u(f_v) \text{ and } \forall l' < l : v_{l',i'} = 0\} \\
\mathit{while}\ l \neq n : \\
\qquad \left.\begin{array}{l}
\{P_u(f_v) \text{ and } \forall l' < l : v_{l',i'} = 0 \text{ and } l \neq n\} \\
i = 1 \\
\mathit{while}\ i \neq 2^l : \\
\qquad v_{l+1,2i-1} = v_{l+1,2i-1} + v_{l,i}/2 \\
\qquad v_{l+1,2i} = v_{l+1,2i} + v_{l,i} \\
\qquad v_{l+1,2i+1} = v_{l+1,2i+1} + v_{l,i}/2 \\
\qquad v_{l,i} = 0 \\
\qquad i = i + 1 \\
l = l + 1 \\
\{P_u(f_v) \text{ and } \forall l' < l : v_{l',i'} = 0\}
\end{array}\right\} \quad \text{proposition} \\
\{P_u(f_v) \text{ and } \forall l' < n : v_{l',i'} = 0\} \quad \text{(follows)}
\end{array}
$$

Proof: The inner loop

$$X := \forall l' < l : v_{l',i'} = 0 \text{ and } \forall i' < i : v_{l,i'} = 0 \text{ and } l \neq n$$

$\{P_u(f_v) \text{ and } \forall l' < l : v_{l',i'} = 0 \text{ and } l \neq n\}$
$i = 1$
$\{P_u(f_v) \text{ and } X\}$
while $i \neq 2^l$:

$$\left.\begin{array}{l} \{P_u(f_v) \text{ and } X \text{ and } i \neq 2^l\} \\ v_{l+1,2i-1} = v_{l+1,2i-1} + v_{l,i}/2 \\ v_{l+1,2i} = v_{l+1,2i} + v_{l,i} \\ v_{l+1,2i+1} = v_{l+1,2i+1} + v_{l,i}/2 \\ v_{l,i} = 0 \\ i = i + 1 \\ \{P_u(f_v) \text{ and } X\} \end{array}\right\} \text{proposition}$$

$\{P_u(f_v) \text{ and } \forall l' < l : v_{l',i'} = 0 \text{ and } \forall i' < 2^l : v_{l,i'} = 0\}$ (follows)
$l = l + 1$
$\{P_u(f_v) \text{ and } \forall l' < l : v_{l',i'} = 0\}$ (follows)

Proof: End

The following assertions are equivalent:

- $\{P_u(f_v)$ and $X\}$
- $\{P_u(f_v + v_{l,i}\underbrace{(-\Phi_{l,i} + \Phi_{l+1,2i+1}/2 + \Phi_{l+1,2i} + \Phi_{l+1,2i-1}/2)}_{=0})$ and $X\}$

The algorithm terminates in all cases because it consists only of for loops.

qed

Representation in the hierarchical basis

How can we represent the function given by the vector v in the hierarchical basis?

Algorithm `toHierarchicalBasis`:

- input
  - integer $n > 1$
  - vector v with $\sum_{l=1}^{n} \sum_{i=1}^{2^l-1} v_{l,i} \Phi_{l,i} = u$
- output
  - vector v with $\sum_{l=1}^{n} \sum_{i=1}^{2^l-1} v_{l,i} \Phi_{l,i} = u$ and $v_{l,i} = 0$ for all even i

Conversion into the hierarchical basis

Algorithm for conversion of a vector v into the hierarchical basis:

toHierarchicalBasis (wrong!)

$$
\begin{array}{l}
\textit{for} \quad l = n-1, \ldots, 1: \\
\quad \textit{for} \quad i = 1, \ldots, 2^l - 1: \\
\quad\quad v_{l+1,2i-1} \mathrel{-}= v_{l+1,2i}/2 \\
\quad\quad v_{l+1,2i+1} \mathrel{-}= v_{l+1,2i}/2 \\
\quad\quad v_{l,i} = v_{l+1,2i} \\
\quad\quad v_{l+1,2i} = 0
\end{array}
$$

In the following we prove the correctness of the (corrected) algorithm.

Transformation of the program

We have to transform our program so that it uses only the control structures covered by the Hoare logic:

toHierarchicalBasis (wrong!)

$l = n - 1$
$\textit{while}\ l \neq 0:$
$\quad i = 1$
$\quad \textit{while}\ i \neq 2^l:$
$\quad\quad v_{l+1,2i-1} = v_{l+1,2i-1} - v_{l+1,2i}/2$
$\quad\quad v_{l+1,2i+1} = v_{l+1,2i+1} - v_{l+1,2i}/2$
$\quad\quad v_{l,i} = v_{l+1,2i}$
$\quad\quad v_{l+1,2i} = 0$
$\quad\quad i = i + 1$
$\quad l = l - 1$

Proof: The outer loop

$$
\begin{array}{l}
\{P_u(f_v)\} \\
l = n - 1 \\
\{P_u(f_v) \text{ and } \forall l' > l+1 : v_{l',2i'} = 0\} \\
\textit{while } l \neq 0 : \\
\left.\begin{array}{l}
\quad \{P_u(f_v) \text{ and } \forall l' > l+1 : v_{l',2i'} = 0 \text{ and } l \neq 0\} \\
\quad i = 1 \\
\quad \textit{while } i \neq 2^l : \\
\qquad v_{l+1,2i-1} = v_{l+1,2i-1} - v_{l+1,2i}/2 \\
\qquad v_{l+1,2i+1} = v_{l+1,2i+1} - v_{l+1,2i}/2 \\
\qquad v_{l,i} = v_{l+1,2i} \\
\qquad v_{l+1,2i} = 0 \\
\qquad i = i + 1 \\
\quad l = l - 1 \\
\quad \{P_u(f_v) \text{ and } \forall l' > l+1 : v_{l',2i'} = 0\}
\end{array}\right\} \text{proposition} \\
\{P_u(f_v) \text{ and } \forall l' > 1 : v_{l',2i'=0}\} \text{ (follows)}
\end{array}
$$

Proof: The inner loop

$$X := \forall l' > l+1 : v_{l',2i'} = 0 \text{ and } \forall i' < i : v_{l+1,2i'} = 0 \text{ and } l \neq 0$$

$\{P_u(f_v) \text{ and } \forall l' > l+1 : v_{l',2i'} = 0 \text{ and } l \neq 0\}$
$i = 1$
$\{P_u(f_v) \text{ and } X\}$
$\mathit{while}\ i \neq 2^l :$

$$\left.\begin{array}{l} \{P_u(f_v) \text{ and } X\} \\ v_{l+1,2i-1} = v_{l+1,2i-1} - v_{l+1,2i}/2 \\ v_{l+1,2i+1} = v_{l+1,2i+1} - v_{l+1,2i}/2 \\ v_{l,i} = v_{l+1,2i} \\ v_{l+1,2i} = 0 \\ i = i+1 \\ \{P_u(f_v) \text{ and } X\} \end{array}\right\} \text{proposition}$$

$\{P_u(f_v) \text{ and } \forall l' > l : v_{l',2i'} = 0\}$ (follows)
$l = l - 1$
$\{P_u(f_v) \text{ and } \forall l' > l+1 : v_{l',2i'} = 0\}$ (follows)

Proof: The inner block (1)

$$X := \forall l' > l+1 : v_{l',2i'} = 0 \text{ and } \forall i' < i : v_{l+1,2i'} = 0 \text{ and } l \neq 0$$

$\{P_u(f_v) \text{ and } X\}$
$\{P_u(f_v + v_{l+1,2i}(\Phi_{l,i} - \Phi_{l+1,2i} - \Phi_{l+1,2i+1}/2 - \Phi_{l+1,2i-1}/2) - v_{l,i}\Phi_{l,i}) \text{ and } X\}$
$v_{l+1,2i-1} = v_{l+1,2i-1} - v_{l+1,2i}/2$
$\{P_u(f_v + v_{l+1,2i}(\Phi_{l,i} - \Phi_{l+1,2i} - \Phi_{l+1,2i+1}/2) - v_{l,i}\Phi_{l,i}) \text{ and } X\}$
$v_{l+1,2i+1} = v_{l+1,2i+1} - v_{l+1,2i}/2$
$\{P_u(f_v + v_{l+1,2i}(\Phi_{l,i} - \Phi_{l+1,2i}) - v_{l,i}\Phi_{l,i}) \text{ and } X\}$
$v_{l,i} = v_{l+1,2i}$
$\{P_u(f_v - v_{l+1,2i}\Phi_{l+1,2i}) \text{ and } X\}$
$v_{l+1,2i} = 0$
$\{P_u(f_v) \text{ and } X \text{ and } v_{l+1,2i} = 0\}$
$i = i + 1$
$\{P_u(f_v) \text{ and } X\}$

Proof: The inner block (2)

We would have to show that the following assertions are equivalent:

- $\{P_u(f_v)$ and $X\}$
- $\{P_u(f_v + v_{l+1,2i}(\Phi_{l,i} - \Phi_{l+1,2i} - \Phi_{l+1,2i+1}/2 - \Phi_{l+1,2i-1}/2) - v_{l,i}\Phi_{l,i})$ and $X\}$

We already know that

$$\Phi_{l,i} = \frac{\Phi_{l+1,2i-1}}{2} + \Phi_{l+1,2i} + \frac{\Phi_{l+1,2i+1}}{2}$$

So something is wrong with $-v_{l,i}\Phi_{l,i}$.

Proof: The inner block (corrected)

$\{P_u(f_v) \text{ and } X\}$
$\{P_u(f_v + v_{l+1,2i}(\Phi_{l,i} - \Phi_{l+1,2i} - \Phi_{l+1,2i+1}/2 - \Phi_{l+1,2i-1}/2))$
$\text{ and } X\}$
$v_{l+1,2i-1} = v_{l+1,2i-1} - v_{l+1,2i}/2$
$\{P_u(f_v + v_{l+1,2i}(\Phi_{l,i} - \Phi_{l+1,2i} - \Phi_{l+1,2i+1}/2)) \text{ and } X\}$
$v_{l+1,2i+1} = v_{l+1,2i+1} - v_{l+1,2i}/2$
$\{P_u(f_v + v_{l+1,2i}(\Phi_{l,i} - \Phi_{l+1,2i})) \text{ and } X\}$
$v_{l,i} = v_{l,i} + v_{l+1,2i}$
$\{P_u(f_v - v_{l+1,2i}\Phi_{l+1,2i}) \text{ and } < X\}$
$v_{l+1,2i} = 0$
$\{P_u(f_v) \text{ and } X \text{ and } v_{l+1,2i} = 0\}$
$i = i + 1$
$\{P_u(f_v) \text{ and } X\}$

qed

Coversion into the the hierarchical basis (corrected)

So we have the following corrected algorithm for converting a vector v into the hierarchical basis:

toHierarchicalBasis

$$
\begin{aligned}
&\textit{for}\quad l = n-1, \ldots, 1: \\
&\qquad \textit{for}\quad i = 1, \ldots, 2^l - 1: \\
&\qquad\qquad v_{l+1,2i-1} \mathrel{-}= v_{l+1,2i}/2 \\
&\qquad\qquad v_{l+1,2i+1} \mathrel{-}= v_{l+1,2i}/2 \\
&\qquad\qquad v_{l,i} \mathrel{+}= v_{l+1,2i} \\
&\qquad\qquad v_{l+1,2i} = 0
\end{aligned}
$$

Obviously the algorithm terminates in all cases :)

Conclusion

We have seen:

- How we can prove the correctness of an algorithm
- How we can find bugs with the Hoare logic

This presentation is based on a presentation by Samuel Kerschbaumer.