

# Adaptieve benaderingsmethodes

Jan Westerdiep

7 april 2014

# Wat?

- Adaptieve benaderingsmethodes

# Wat?

- Adaptieve benaderingsmethodes
- Functies benaderen

# Wat?

- **Adaptieve** benaderingsmethodes
- Functies benaderen
- Door te kijken *waar* de functie moeilijk is en daar in te zoomen

# Wat?

- Adaptieve benaderingsmethodes
- Functies benaderen
- Door te kijken *waar* de functie moeilijk is en daar in te zoomen
- Dit alles aan de hand van twee artikelen van Peter Binev

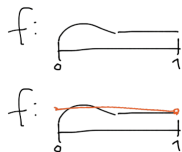
## 2004 – iets preciezer

- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein en laat  $n \in \mathbb{N}$  vast



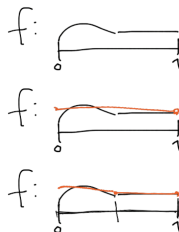
## 2004 – iets preciezer

- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein en laat  $n \in \mathbb{N}$  vast
- Op elk stukje, best benaderende polynoom van graad  $n$



## 2004 – iets preciezer

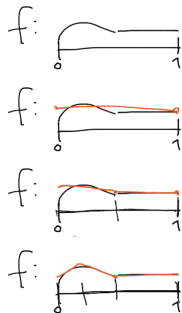
- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein en laat  $n \in \mathbb{N}$  vast
- Op elk stukje, best benaderende polynoom van graad  $n$
- *Kies* stukje om op te delen





## 2004 – iets preciezer

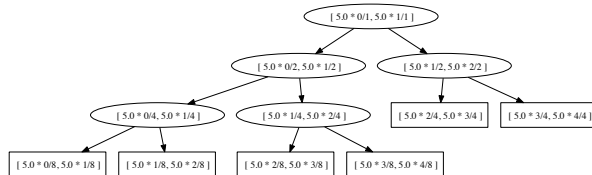
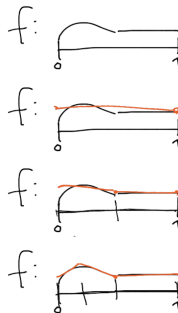
- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein en laat  $n \in \mathbb{N}$  vast
- Op elk stukje, best benaderende polynoom van graad  $n$
- Kies stukje om op te delen
- Herhaal



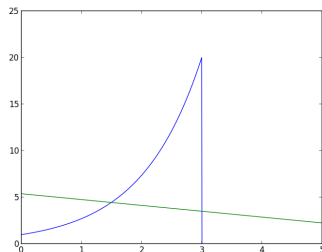
# 2004 – iets preciezer

- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein en laat  $n \in \mathbb{N}$  vast
- Op elk stukje, best benaderende polynoom van graad  $n$
- Kies stukje om op te delen
- Herhaal

⇒ Tree Generating Algorithm

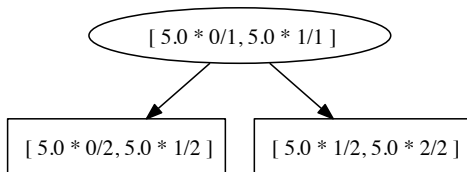
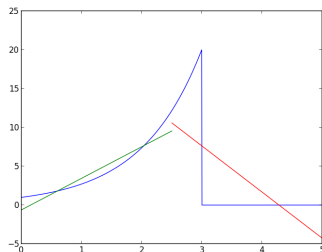


# Enkel door midden hakken

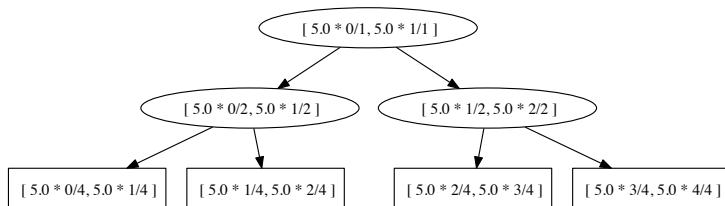
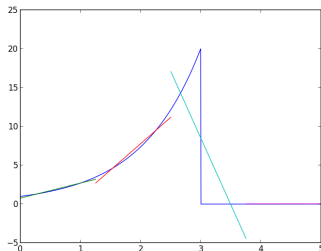


[  $5.0 * 0/1$ ,  $5.0 * 1/1$  ]

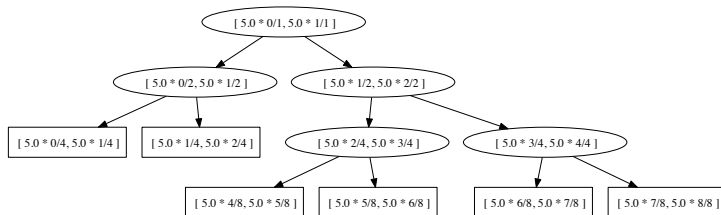
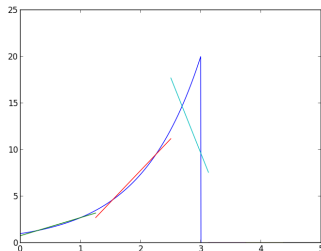
# Enkel door midden hakken



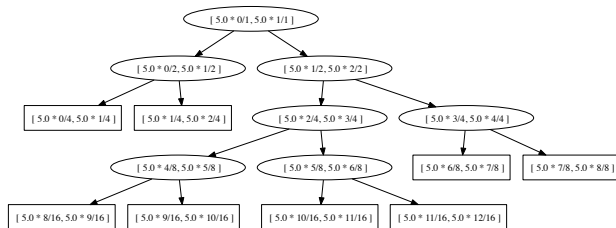
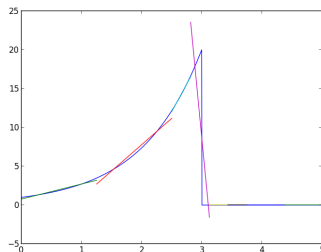
# Enkel door midden hakken



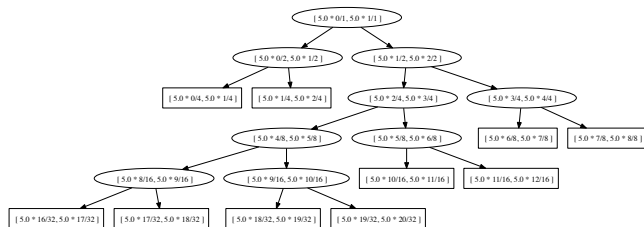
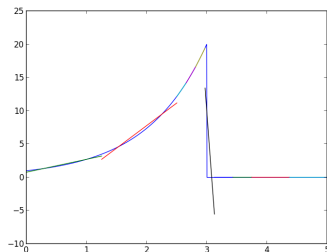
# Enkel door midden hakken



# Enkel door midden hakken

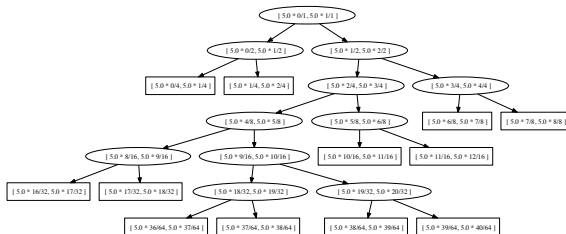
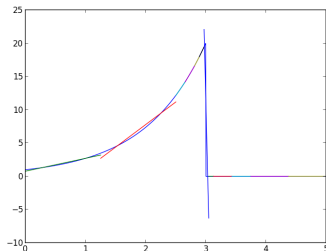


# Enkel door midden hakken

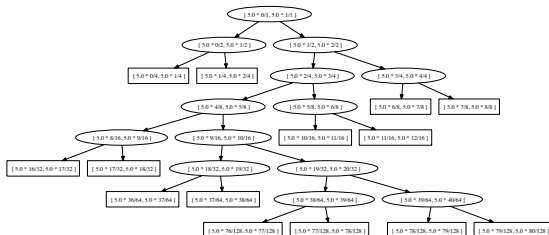
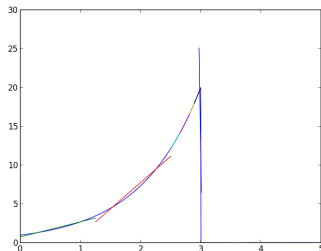




# Enkel door midden hakken



## Enkel door midden hakken



## 2004 – iets preciezer

- Bewijsbaar: Binev-2004 maakt een boom die na  $n$  opdelingen
  - gevonden is in  $\mathcal{O}(n)$ , en
  - beter is dan de optimale boom in  $n/2$  opdelingen

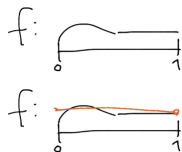
## 2013 – iets preciezer

- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein



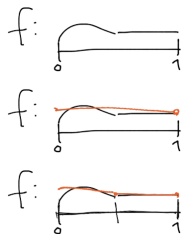
# 2013 – iets preciezer

- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein
- Op elk stukje, best benaderende polynoom van bepaalde graad



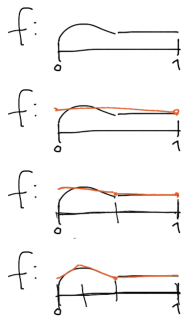
# 2013 – iets preciezer

- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein
- Op elk stukje, best benaderende polynoom van bepaalde graad
- Kies stukje om op te delen



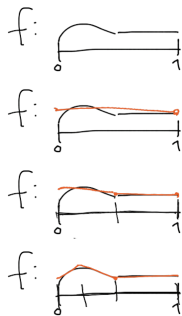
# 2013 – iets preciezer

- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein
- Op elk stukje, best benaderende polynoom van bepaalde graad
- Kies stukje om op te delen
- Herhaal



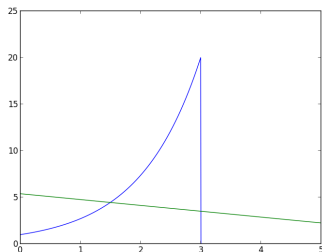
# 2013 – iets preciezer

- Pak  $f : [a, b] \rightarrow \mathbb{R}$  met een opdeling van het domein
  - Op elk stukje, best benaderende polynoom van bepaalde graad
  - Kies stukje om op te delen
- ⇒ Of verhoog de graad van het polynoom
- Herhaal





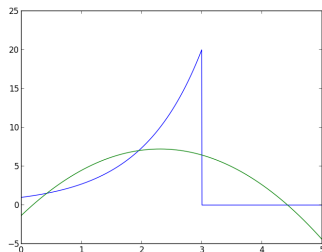
## Ook polynomiale graad verhogen



$$\begin{bmatrix} 5.0 * 0/1, 5.0 * 1/1 \end{bmatrix}$$

$p=2$

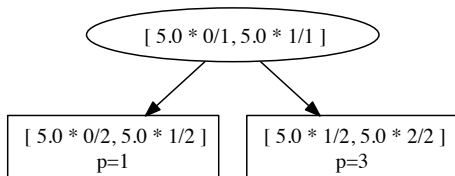
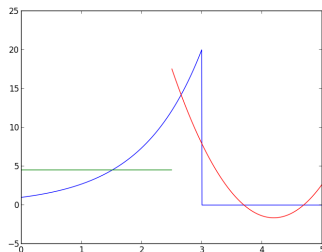
## Ook polynomiale graad verhogen



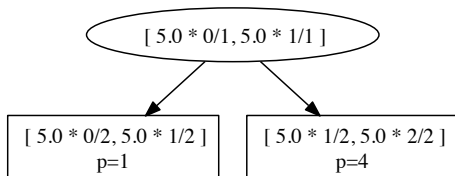
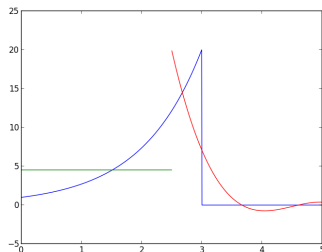
$$\begin{bmatrix} 5.0 * 0/1, 5.0 * 1/1 \end{bmatrix}$$

$p=3$

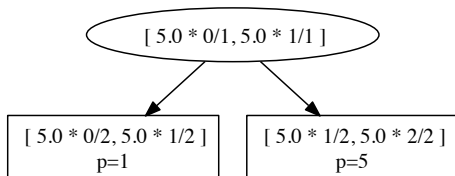
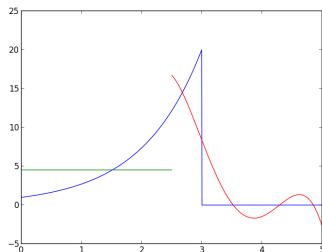
# Ook polynomiale graad verhogen



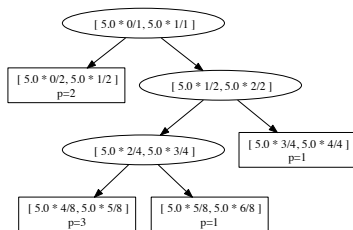
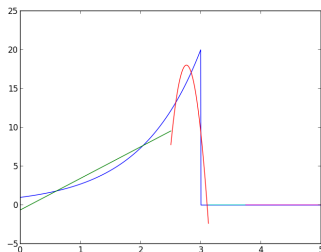
# Ook polynomiale graad verhogen



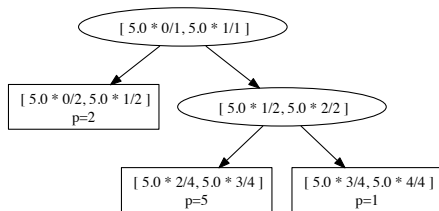
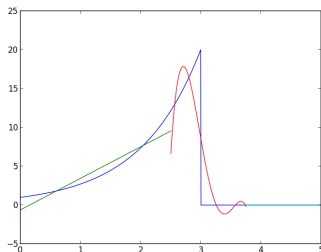
# Ook polynomiale graad verhogen



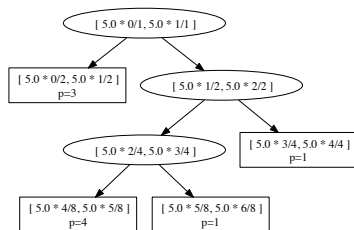
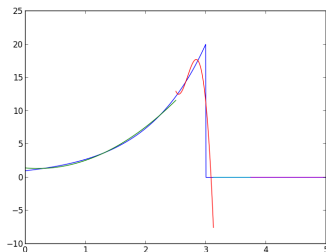
# Ook polynomiale graad verhogen



# Ook polynomiale graad verhogen

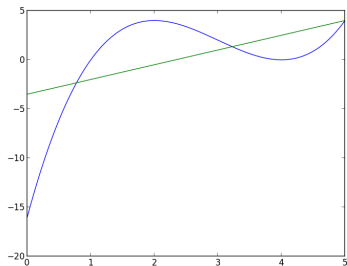
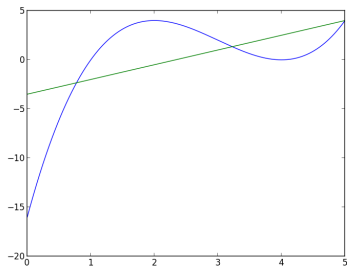


# Ook polynomiale graad verhogen

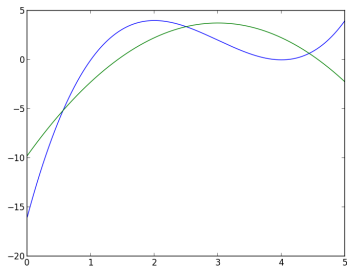
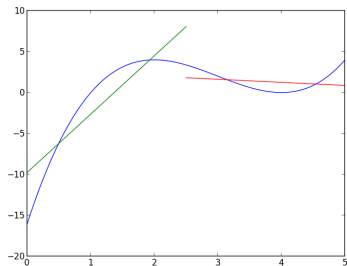




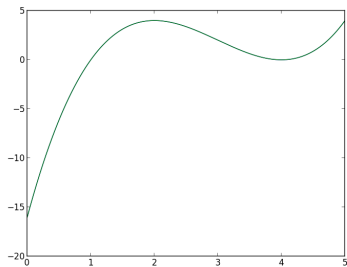
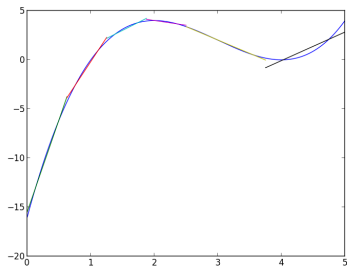
## Ander voorbeeld



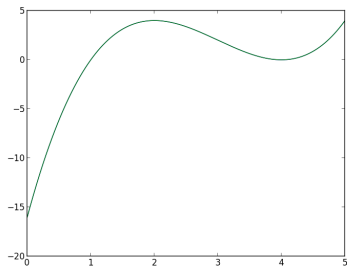
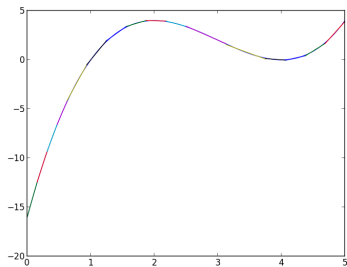
## Ander voorbeeld



## Ander voorbeeld



## Ander voorbeeld



# Bij wie?

Rob Stevenson

- Wiskunde – Theorie



# Bij wie?

Rob Stevenson

- Wiskunde – Theorie

Dick van Albada

- Informatica – Praktijk



# Waarom?

- Wiskunde op de computer: goede samenkomst wiskunde/informatica

# Waarom?

- Wiskunde op de computer: goede samenkomst  
wiskunde/informatica
- Spiksplinternieuw (2013)



# Waarom?

- Wiskunde op de computer: goede samenkomst wiskunde/informatica
- Spiksplinternieuw (2013)
- Vraagstelling is makkelijk, de theorie erachter niet

# Waarom?

- Wiskunde op de computer: goede samenkomst wiskunde/informatica
- Spiksplinternieuw (2013)
- Vraagstelling is makkelijk, de theorie erachter niet
- Veel uitbreiding en toepassing mogelijk

# Wanneer?

- Februari, maart, april: Theorie begrijpen, implementatie in Python

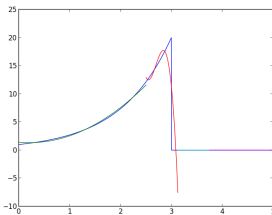
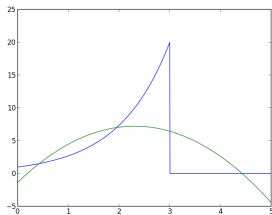
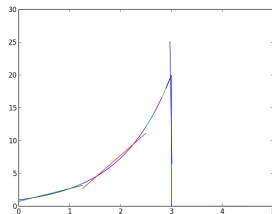
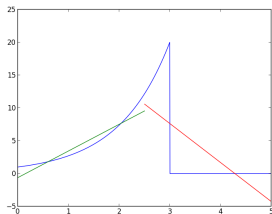
# Wanneer?

- Februari, maart, april: Theorie begrijpen, implementatie in Python
- Mei: Implementatie in C

# Wanneer?

- Februari, maart, april: Theorie begrijpen, implementatie in Python
- Mei: Implementatie in C
- Daarna: High Performance, paralleliseren, toepassen

# Jan Westerdiep – Adaptieve benaderingsmethodes



# Schets van inhoudsopgave

## 1 1-dimensionale geval

# Schets van inhoudsopgave

- 1 1-dimensionale geval
- 2 2-dimensionale geval



# Schets van inhoudsopgave

- 1 1-dimensionale geval
- 2 2-dimensionale geval
- 3 Implementatie in C

# Schets van inhoudsopgave

- 1 1-dimensionale geval
- 2 2-dimensionale geval
- 3 Implementatie in C
- 4 Parallelliseren

# Schets van inhoudsopgave

- 1 1-dimensionale geval
- 2 2-dimensionale geval
- 3 Implementatie in C
- 4 Parallelliseren
- 5 Implementatie in e.o.a. parallelle situatie (GPU, supercomputer)

# Schets van inhoudsopgave

- 1 1-dimensionale geval
- 2 2-dimensionale geval
- 3 Implementatie in C
- 4 Parallelliseren
- 5 Implementatie in e.o.a. parallelle situatie (GPU, supercomputer)
- 6 (wellicht) Uitbreiding en toepassing