

# Numerische Quadratur

Patrick Radner, Jennifer Sorensen, Florian Zopf

Aspekte der systemnahen Programmierung bei der Spieleentwicklung  
Projektaufgabe - 31

16.04.2015

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

## Theorie

Numerische Quadratur  
Newton-Cotes Formeln  
Aufgabenstellung

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions für Fließkommazahlen  
Dynamische Bibliotheken  
Assembler Implementierung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

## Theorie

Numerische Quadratur  
Newton-Cotes Formeln  
Aufgabenstellung

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions für Fließkommazahlen  
Dynamische Bibliotheken  
Assembler Implementierung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

- ▶ Lösung eines mathematischen Problems
- ▶ Annähern von Integralen durch Summation von Flächen
- ▶ Newton-Cotes Formeln

## Theorie

### Numerische Quadratur

Newton-Cotes  
Formeln

Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln

Calling Conventions  
für Fließkommazahlen

Dynamische  
Bibliotheken

Assembler  
Implementierung

## Evaluierung

Performance

Genauigkeit

Security-Bedenken

$$I_n(p) = \int_a^b p(x) dx = (b-a) * \sum_{i=0}^n \alpha_{i,n} * p(x_i)$$

- ▶ Stützstellen:

$$x_i = a + \frac{i * (b-a)}{n}$$

- ▶ Gewichte:

$$\alpha_{i,n} = \frac{1}{n} * \int_0^n \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-j}{x-i} dx$$

## Theorie

Numerische  
Quadratur

Newton-Cotes  
Formeln

Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln

Calling Conventions  
für Fließkommazahlen

Dynamische  
Bibliotheken

Assembler  
Implementierung

## Evaluierung

Performance

Genauigkeit

Security-Bedenken

- ▶ NC-Formeln für  $n=1,2$

- ▶ ASM-Funktionen (nur Grundrechenarten):

```
float *newton_cotes_1(float (*p)(float), int a, int b)
```

```
double *newton_cotes_2(double (*p)(double), int a, int b)
```

- ▶ Polynom aus dynamischer Bibliothek

```
double f(double x)
```

- ▶ Evaluierung: Genauigkeit und Performance
- ▶ Security-Bedenken?

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln

## Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

## Theorie

Numerische Quadratur  
Newton-Cotes Formeln  
Aufgabenstellung

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions für Fließkommazahlen  
Dynamische Bibliotheken  
Assembler Implementierung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

# Vereinfachte Newton-Cotes Formeln

$$I_1(p) = \frac{1}{2} * (b - a) * (p(a) + p(b))$$

$$I_2(p) = \frac{1}{6} * (b - a) * (p(a) + 4 * p(\frac{a+b}{2}) + p(b))$$

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

**Vereinfachte Formeln**  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

# Calling Conventions für Fließkommazahlen

Numerische  
Quadratur

Patrick Radner,  
Jennifer Sorensen,  
Florian Zopf

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
**Calling Conventions  
für Fließkommazahlen**  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

- ▶ *-mfloat-abi=soft/softfp/hard*
- ▶ *hard: d0-d3* als Übergaberegister

▶ POSIX-Aufrufe:

```
void *dlopen(const char *filename, int flag);  
void *dlsym(void *handle, const char *symbol);
```

▶ `-ldl` beim Compilieren

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
**Dynamische  
Bibliotheken**  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

- ▶ Vereinfachte Formeln
- ▶ Parameter:
  - ▶ r0: Funktionspointer
  - ▶ r1,r2: a,b
- ▶ Return: d0
- ▶ Funktionsaufruf → saved Registers

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken

## Assembler Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

# Übersicht

## Theorie

Numerische Quadratur  
Newton-Cotes Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions für Fließkommazahlen  
Dynamische Bibliotheken  
Assembler Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

### Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

### Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

### Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

- ▶ Vereinfachte NC-Formeln lassen wenig Platz für Optimierung
- ▶ Assembler und C-Implementierung unterscheiden sich kaum.
- ▶ 1. Funktion ca. 2ms langsamer
- ▶ 2. Funktion ca. 8ms langsamer

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

**Performance**  
Genauigkeit  
Security-Bedenken

- ▶ Vereinfachte NC-Formeln lassen wenig Platz für Optimierung
- ▶ Assembler und C-Implementierung unterscheiden sich kaum.
- ▶ 1. Funktion ca. 2ms langsamer
- ▶ 2. Funktion ca. 8ms langsamer
- ▶ bei 1 Million Durchläufen.

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

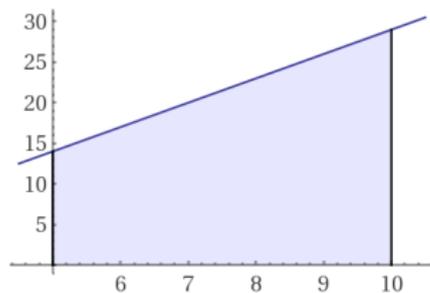
## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

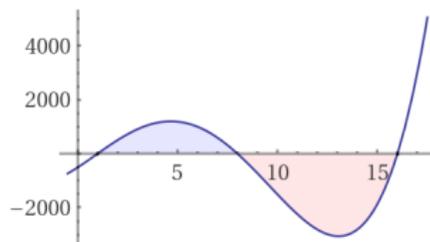
## Evaluierung

**Performance**  
Genauigkeit  
Security-Bedenken

# Genauigkeit der Newton-Cotes Formeln



Computed by Wolfram|Alpha



Computed by Wolfram|Alpha

- ▶ links:  $\int_5^{10} 3x - 1 = 107.5$   
1. und 2. NC-Formel liefern gleichen Wert.
- ▶ rechts:  $\int_1^{16} x^4 - 21x^3 + 52x^2 + 480x - 512 = -9843.8$ 
  - ▶ 1. NC-Formel: 0
  - ▶ 2. NC-Formel: -29882

→ Genauigkeit hängt von der Anzahl der Stützstellen und vom Grad des Polynoms ab.

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
**Genauigkeit**  
Security-Bedenken

- ▶ Nutzer stellt Code zur Verfügung
- ▶ Funktion muss nur passende Signatur haben:

```
double f(double x)
```

- ▶ Kann Schadcode enthalten

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
Security-Bedenken

- ▶ Eingabe des Polynoms via Kommandozeile
- ▶ Entwicklung eines Eingabeformats
- ▶ Parsing benötigt Zeit

## Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

## Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

## Evaluierung

Performance  
Genauigkeit  
**Security-Bedenken**

**Danke für Ihre Aufmerksamkeit!**

**Fragen?**

#### Theorie

Numerische  
Quadratur  
Newton-Cotes  
Formeln  
Aufgabenstellung

#### Lösungsfindung

Vereinfachte Formeln  
Calling Conventions  
für Fließkommazahlen  
Dynamische  
Bibliotheken  
Assembler  
Implementierung

#### Evaluierung

Performance  
Genauigkeit  
**Security-Bedenken**