

## VIII. Zeiger/Pointer

Wdh, VL11 vom 04.07.17

### 1. Motivation und Notation

pointer\_1.cc, pointer\_2.cc, pointer\_swap.cc

- Syntax: `<Typ> * <Name>;`
- \*: Inhaltsoperator  $\Leftrightarrow$  &: Adressoperator
- Nullpointer: `float * ptr; ptr = NULL; oder ptr = 0;`
- Compiler ist streng bei Typenprüfung:  
"Pointer auf int" kann nicht auf "double"-Variable zeigen
- void-Pointer: offen lassen, auf welchen Datentyp ein Pointer zeigt
  - keine Dereferenzierung möglich `pointer_conv.cc, pointer_void.cc`
  - mögliche Anwendung: generische Argumente für Funktionen

### 2. Zeiger, Arrays, Strukturen und Funktionen

#### 2.1 Zeiger und Arrays

Deklaration von Feld `<Typ> a[N]`  $\Leftrightarrow$  Zeigerkonstante `a` definiert.

Deklaration von Pointer `<Typ> * b`  $\Leftrightarrow$  Feld `b[]` steht zur Verfügung.

Pointerarithmetik: `*(a+n) = a[n]` `pointer_array.cc, pointer_an_na.cc, pointer_3.cc`

#### 2.2 Übergabe von Arrays an Funktionen

`void fkt(double v[]){...}` oder `void fkt(double * v){...}` `pointer_4.cc`

#### 2.3 Zeiger auf Funktionen

`pointer_array_fkt.cc`

(a) Funktionspointer: `float (*pfct)(float)`

(b) Fkt. als Parameter: `float integrate(float (*f)(float x))`