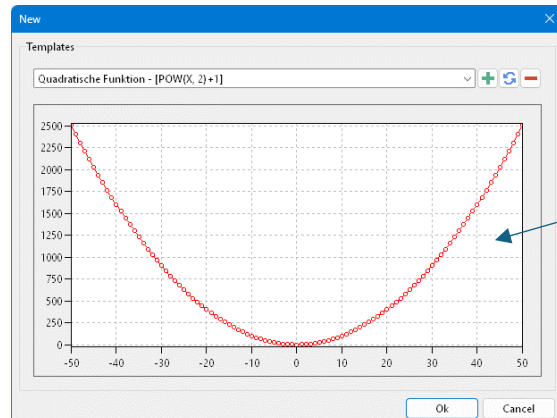
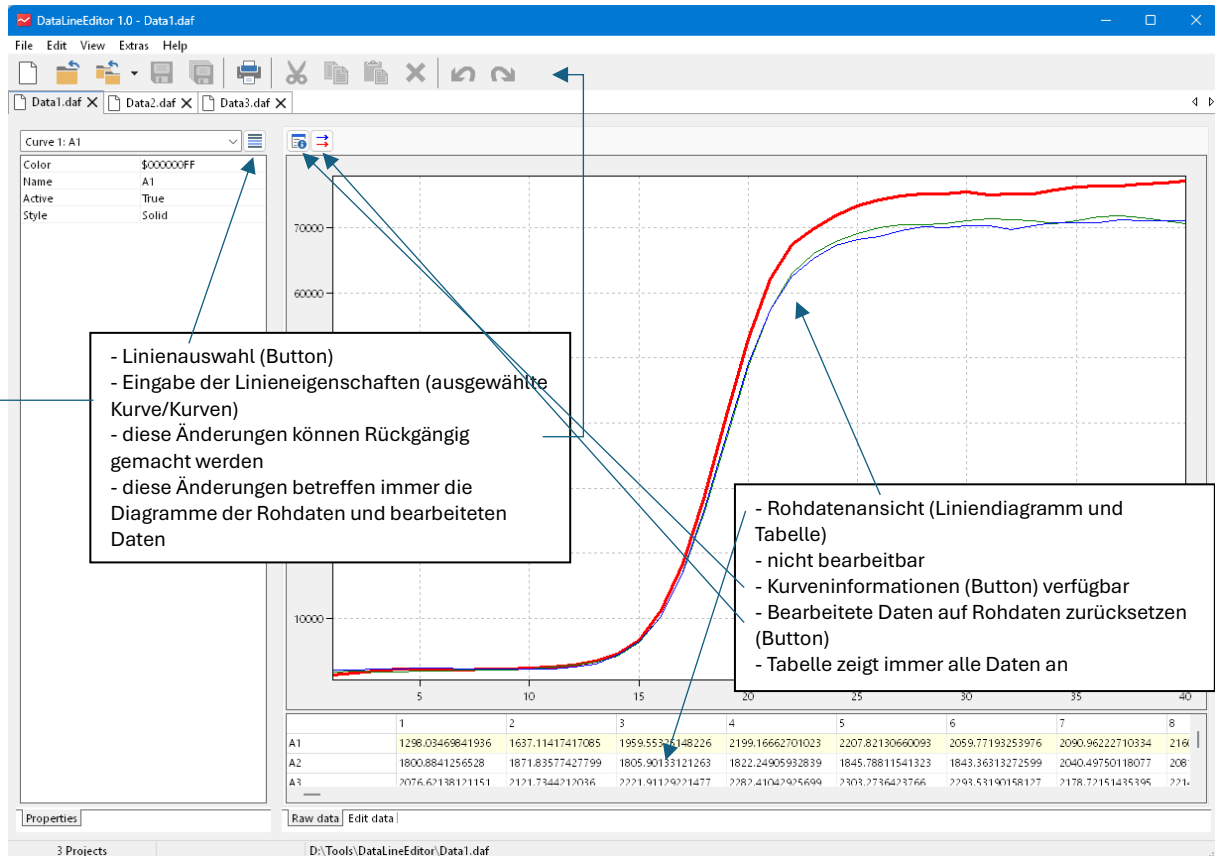
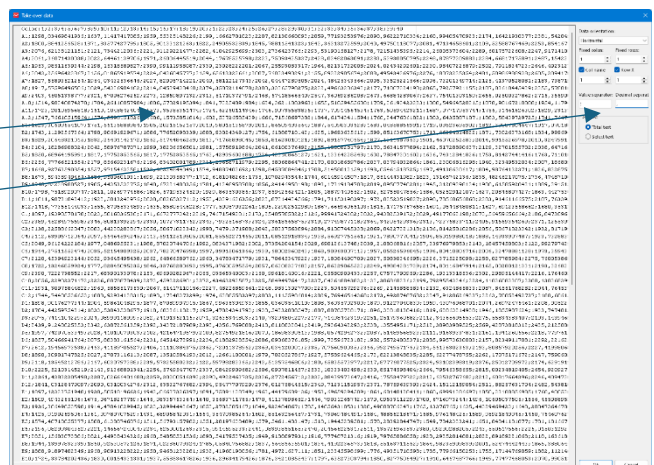


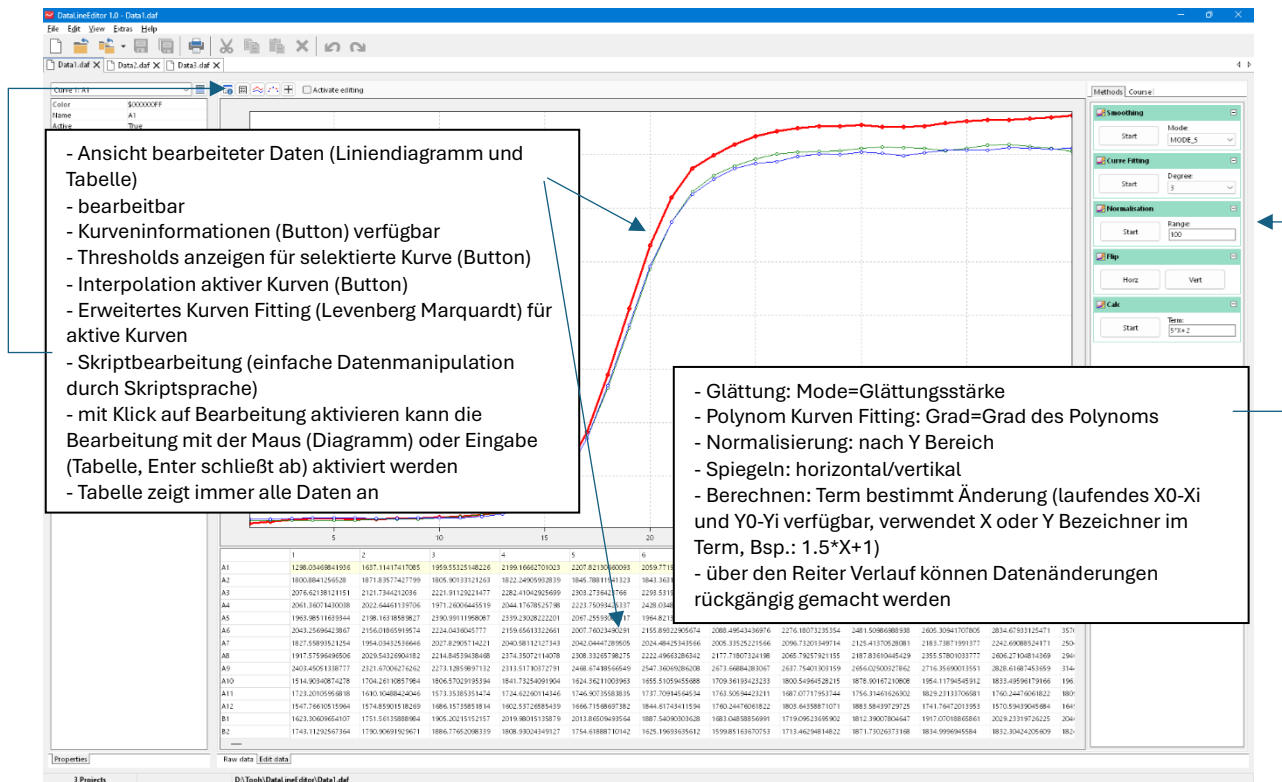
Hauptfenster der Anwendung (Rohdaten)



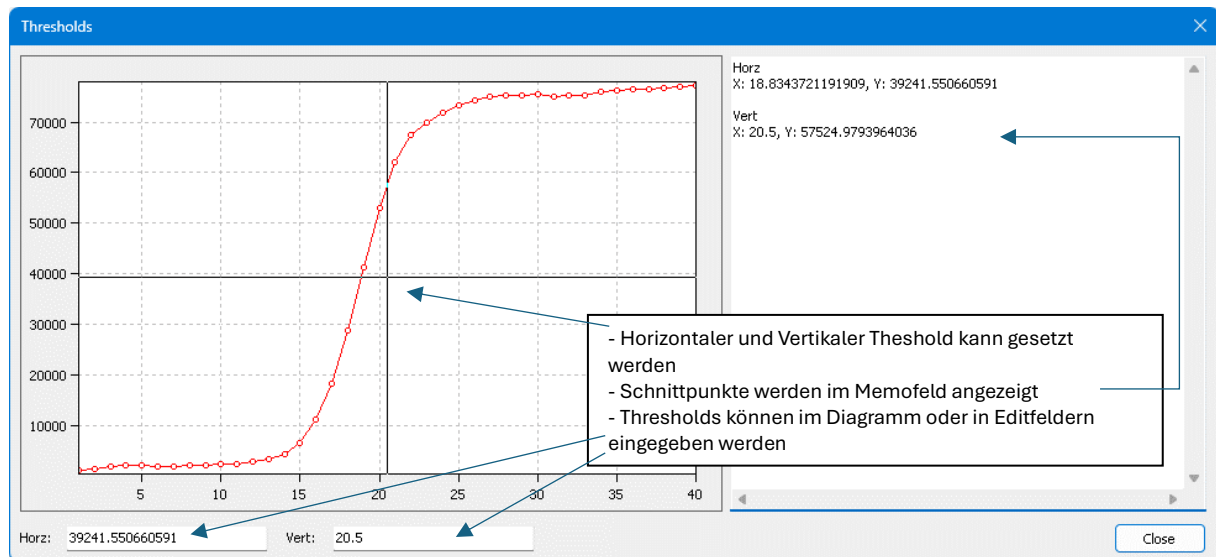
- neues Projekt aus CSV-Datei erstellen
- Daten werden in Textfeld geladen
- Datenorientierung kann horizontal oder vertikal sein
- feste Spalten/Zeilen im Datensatz definieren
- X und Y Zeile/Spalte (je nach Orientierung) innerhalb der festen Spalten/Zeilen festlegen
- Werttrenner und Dezimaltrenner für Datensatz festlegen
- Datenbereich festlegen: Gesamter Text oder Textauswahl



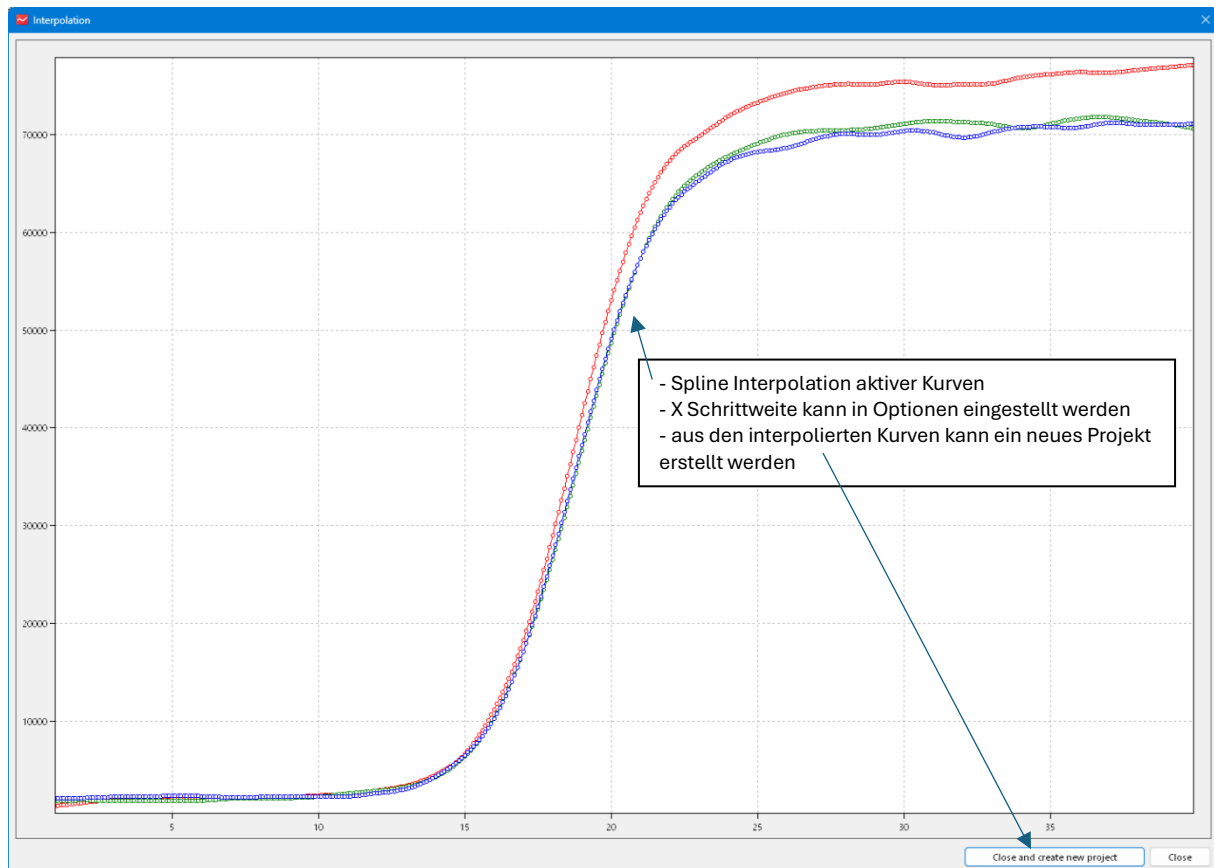
Hauptfenster der Anwendung (bearbeitete Daten)



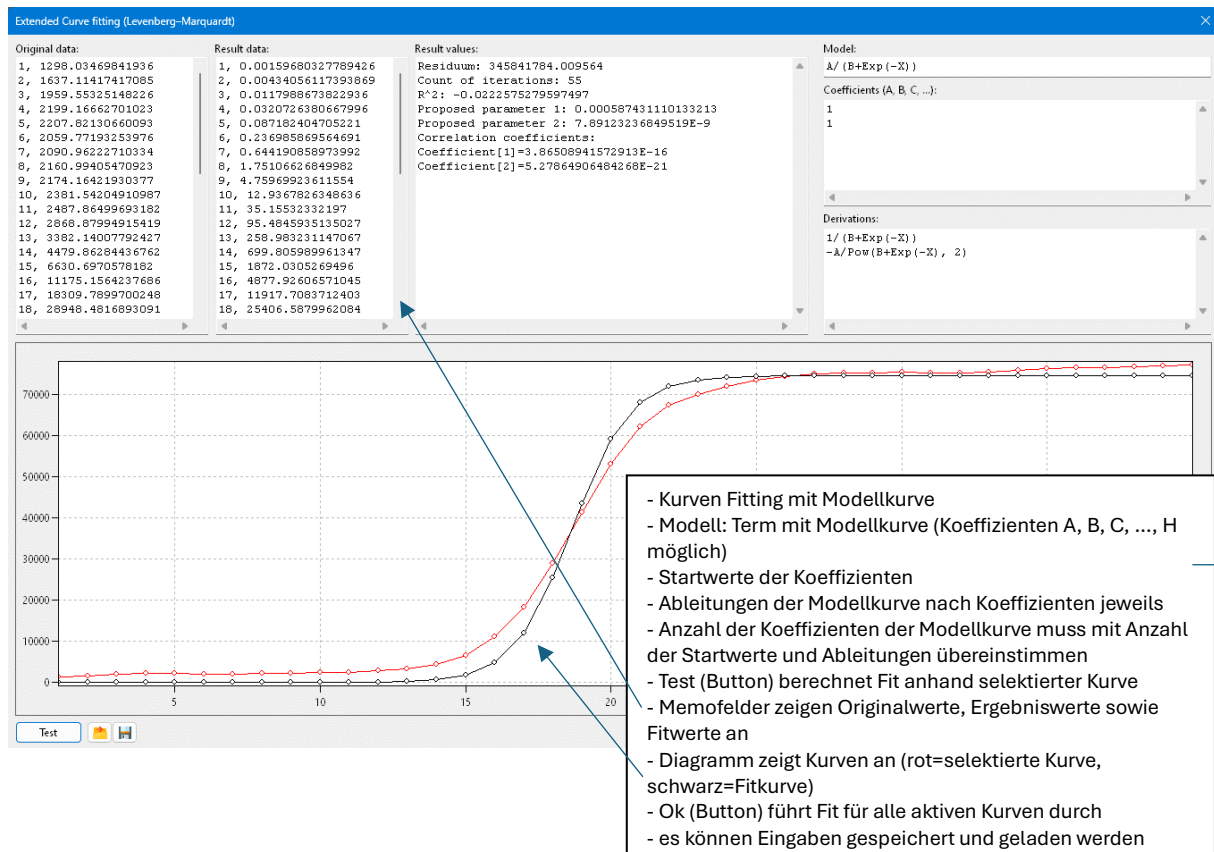
Thresholds setzen und Schnittpunkte bestimmen



Interpolation aktiver Kurven



Erweitertes Kurven Fitting (Levenberg Marquardt)



Programmierung mit einfacher Skriptsprache

Source text:

```
//Glättungsalgorithmus Savitzky
L=getlen(0)
MODE=4
FOR I=0 TO L-1 DO
  B=0
  IF I<MODE THEN
    S=getval(1, I)
    D=setval(2, I, S)
    B=1
  END
  IF I>(L-1)-MODE THEN
    S=getval(1, I)
    D=setval(2, I, S)
    B=2
  END
  IF B=0 THEN
    Sum1=0
    Sum2=0
    FOR J=I-MODE TO I+MODE
      C=array(J-(I-MODE),
        Sum1=Sum1+C*getval(1, J),
        Sum2=Sum2+C;

```

Variables:

```
MODE=4
Sum1=17628543.3466002
Sum2=231
B=2
C=-21
D=77185.0666227626
I=39
J=39
L=40
S=77185.0666227626
```

- Programmierung mit einfacher Skriptsprache (Memo links)
- 3 Arrays stehen für X=0, SrcY=1, DstY=2 zur Verfügung (Zugriff: Punkteanzahl=getlen(ArrayIndex), ArrayWert=getval(ArrayIndex, PunktIndex), Dummy=setval(ArrayIndex, PunktIndex, Wert)
- immer eine Anweisung pro Zeile
- Variable setzen Name=Term (Bsp.: A=5)
- IF Bedingung (Bsp.: A=5 oder A<5 oder A>5, rechts kann auch Term stehen) THEN in einer kommenden Zeile mit END abschließen, ELSE erlaubt
- FOR Bedingung (Bsp.: I=0, rechts kann auch Term stehen) TO Ende (5 oder Term) DO in einer kommenden Zeile mit END abschließen (auch abwärts), BREAK erlaubt
- WHILE Bedingung DO (Bedingung analog IF, analog mit END abschließen), BREAK erlaubt
- immer nur eine Bedingung (IF, FOR, WHILE), also kein and bzw. or
- kein Zeilenabschluss mit Semikolon
- Start (Button) interpretiert Skript anhand selektierter Kurve, Stop beendet
- Memo rechts zeigt Variablenwerte an
- Diagramm zeigt Kurven an (rot=sel. Kurve, schwarz=Skriptkurve DstY)
- es können Eingaben gespeichert und geladen werden

XLS-Excel Export

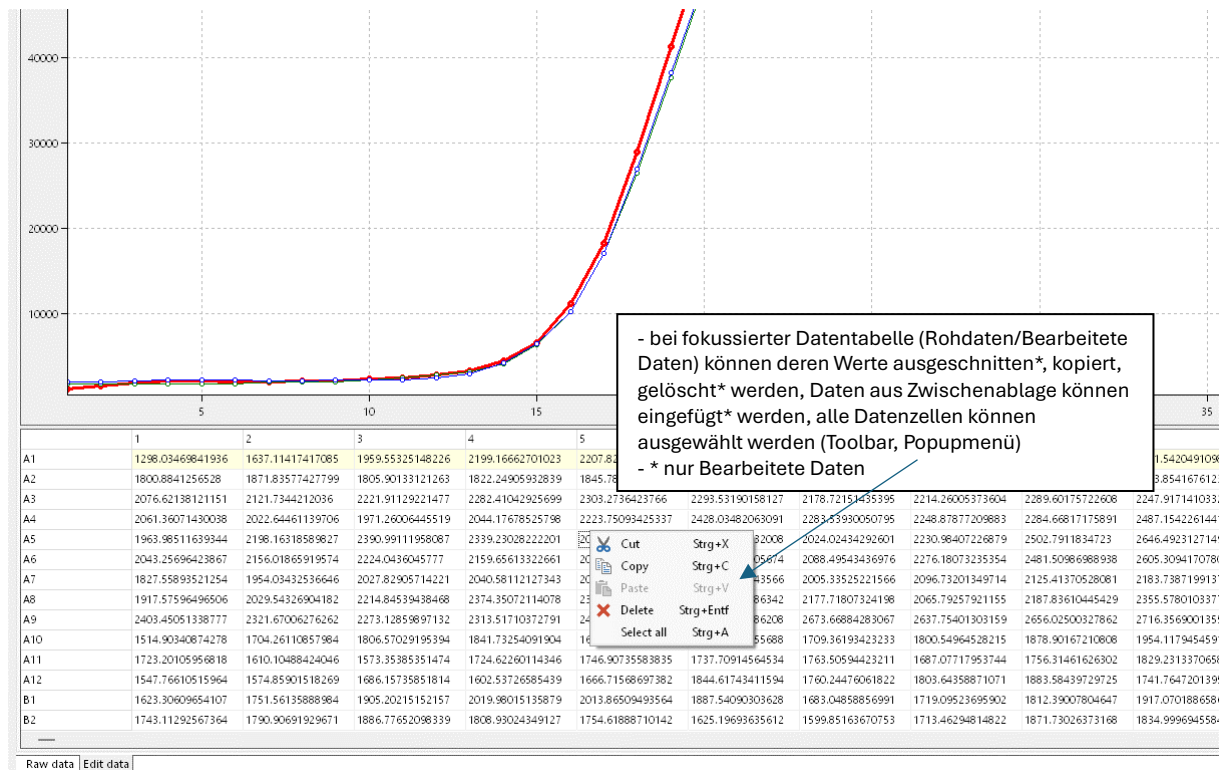
DataLineEditor 1.0 - Data2.daf

File Edit View Extras Help

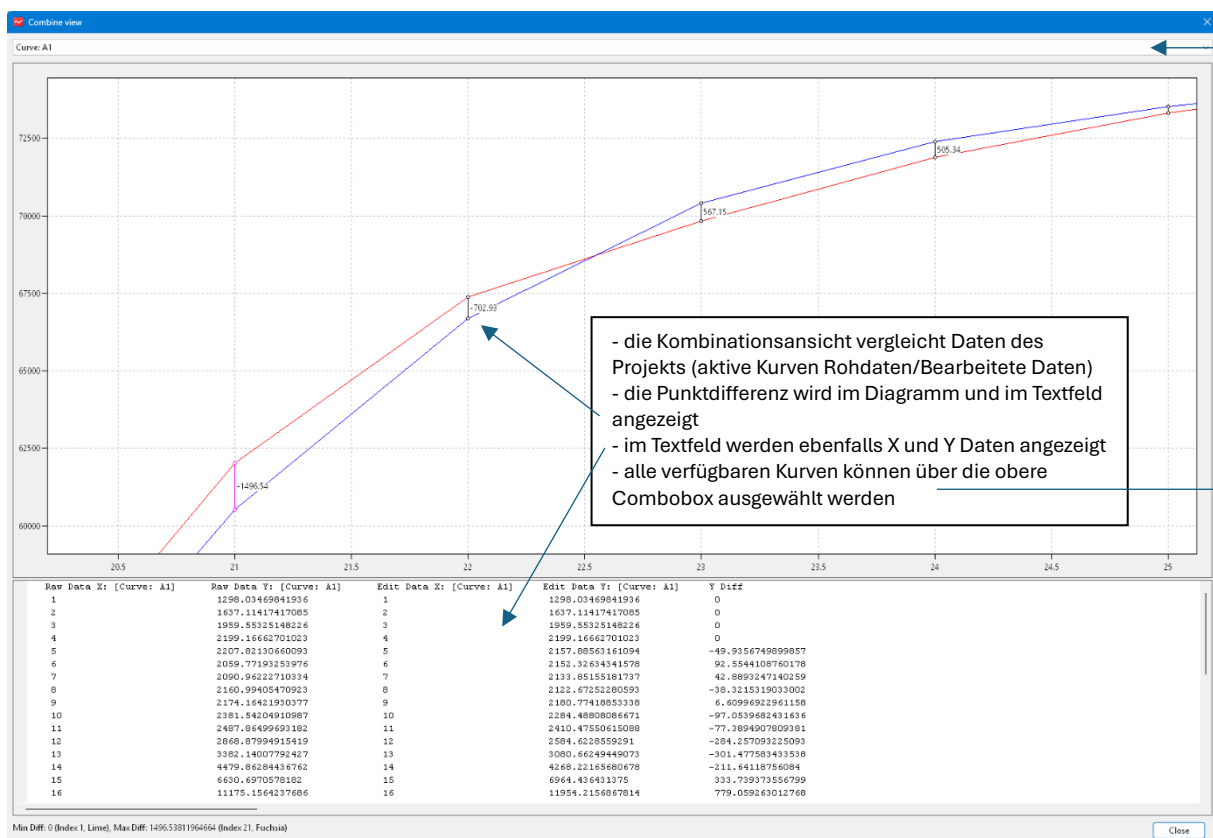
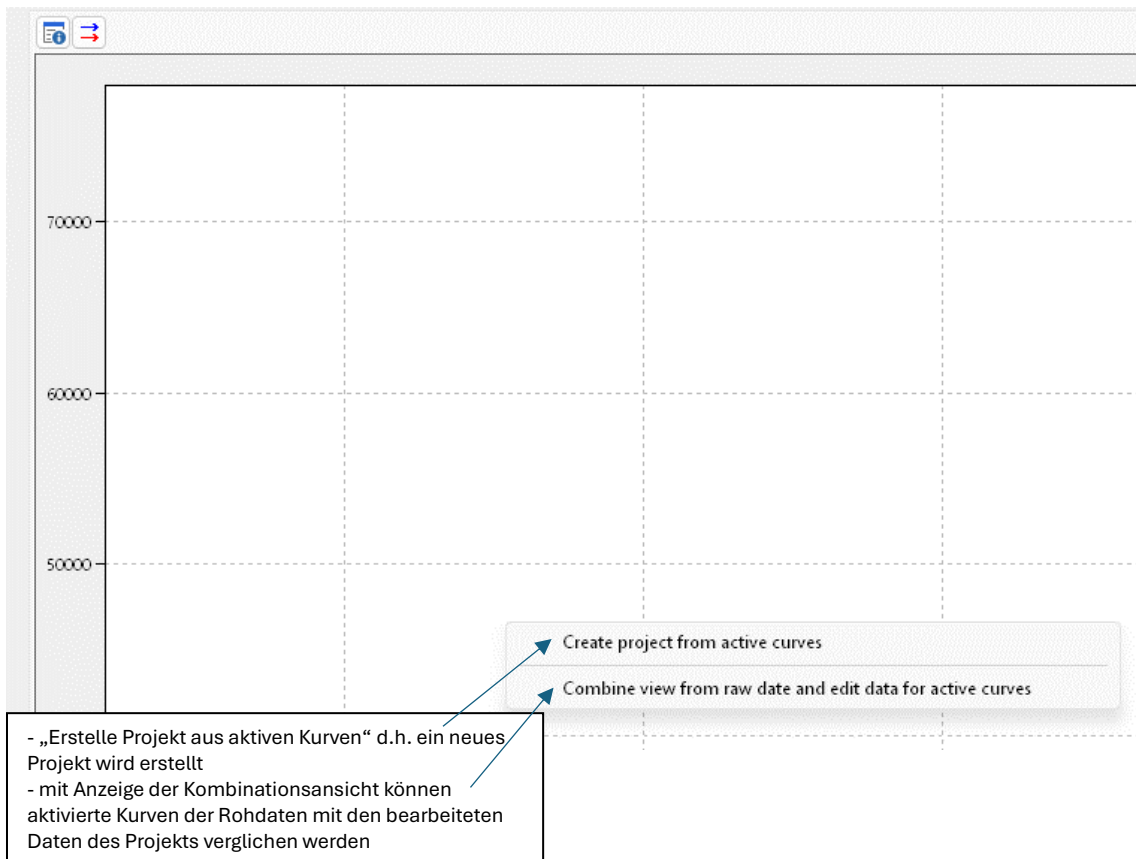
- New Strg+N
- Datei öffnen... Strg+O
- Open project...
- Save project Strg+S
- Save project as...
- Save all projects
- Close project
- Close all projects
- XLS Export...
- Print Strg+P
- Exit

- bei fokussierter Datentabelle (Rohdaten/Bearbeitete Daten) können deren Werte als *.xls gespeichert werden

Ausschneiden, Kopieren, Einfügen, Löschen, Alles markieren



Kombinationsansicht



Hinweise zur Term-Berechnung

- Dezimaltrenner für alle Berechnungen ist der Punkt (.)
- Dezimaltrenner Punkt (.) wird für alle Floatwerte der Anwendung verwendet
- es sind verschiedene Funktionen verfügbar: SIN(..), COS(..), TAN(..), COT(..), ARCSIN(..), ARCCOS(..), ARCTAN(..), ARCCOT(..), POW(.., ..), SQRT(..), MOD(.., ..), DIV(.., ..), ROUND(..), TRUNC(..), LN(..), LOGN(.., ..), EXP(..), ARRAY(Index, Wert1, Wert2, ..), RANDOM(..)
- für die Skriptprogrammierung steht weiterhin zur Verfügung: GETVAL, SETVAL, GETLEN (Beschreibung siehe oben)
- Bsp.1 -> $1.5 \cdot X + 3$
- Bsp.2 -> Quadrat -> $\text{Pow}(5, 2)$
- Bsp.3 -> Quadratwurzel -> $\text{Sqrt}(9)$
- Bsp.4 -> $2 \cdot (5 - (7 + 6))$
- Bsp.5 -> $1 / (7 - 5)$

DataLineEditor 1.0 © 2025 by Thomas Schwobe

Das Programm unterliegt dem Copyright des Autors. Bei erlaubter Weitergabe muss stets die Herkunft angegeben werden. Die Benutzung der Software erfolgt auf eigene Gefahr. Der Autor übernimmt keinerlei Haftung für eventuelle Schäden die durch die Software entstehen. Eine kommerzielle Nutzung ist untersagt.