# Vermessungstechnische Berechnungen

**Programme für Smartphone, Tablet und PC**
**(Android bzw. Windows) auf Python-Basis**
**Version 1.01**

Norbert Fuhrmann, Kerpen




 (Gesetzt in LaTeX)

## Allgemeine Hinweise

Die hier vorliegenden Programme mit vermessungstechnischen Standardaufgaben sind für den Einsatz auf einem Android-Smartphone oder -Tablet sowie für Windows gedacht. Sie können und sollen große vermessungstechnische Programmsysteme mit Datenbankanbindungen nicht ersetzen. Die Programme sind in Python geschrieben worden. Python ist eine höhere, interpretierende Programmiersprache von der <Python Software Foundation>. In Python geschriebener Code lässt sich in diversen Betriebssystemen anwenden. Der Python-Interpreter <IDLE Shell 3.11.0> für Windows kann im Internet unter https://www.python.org/downloads/ heruntergeladen werden. Ein Python-Datei für Windows wie auch für Android hat die Endung *.pyw.

Für die Verwendung in **Windows** gilt (in <Vermessung1-windows.zip>):

Die Datei <Vermessung1.exe> ist ausschließlich für Windows gedacht; dies am besten auf dem Desktop. Sie wurde mit <pyinstaller> Version 5.6.2 aus dem Python-Code erzeugt. **In Windows können mehrere gleiche und/oder unterschiedliche Instanzen geöffnet und nebeneinander benutzt werden.** Die Grafik-Dateien brauchen nicht installiert werden.

Für die Verwendung in **Android** gilt (in <Vermessung1-android.zip>):

Für die Anwendung auf einem Smartphone bzw. Tablet (mit Android) muss im Google <Play store> unter dem Stichwort <python 3 for android> die App <Pydroid 3 - IDE for Python 3>[1] heruntergeladen werden; diese enthält Werbung. Für derzeit € 9,99 kann diese abgeschaltet werden, was für die Leistungsfähigkeit des Programmes hinnehmbar wäre.

Die Programme in Android sind auf eine einzige, sichtbare Instanz hin ausgerichtet (Fullscreen-Modus). Dies ist erforderlich, weil das kleine Display eines Smartphones eine solide Handhabung mit mehreren Fenstern nicht möglich macht.

[1] ⟹ Pydroid_3_-_IDE_for_Python_3.apk (Version 4.01_arm64 bzw. 5.00_arm64).
https://apkpure.com/pydroid-3-ide-for-python-3/ru.iiec.pydroid3/versions
https://pydroid-3-ide-for-python-3.de.softonic.com/android
https://apkfab.com/de/pydroid-3-ide-for-python-3/ru.iiec.pydroid3
Andere, auch neuere Versionen können mitunter Dateipfade nicht finden oder verändern. (Ein allgemeines Problem bei Android; was auch bei Dateimanager auftritt.)

Die gepackte Datei <Vermessung1-android.zip> enthält die Dateien <Vermessung1.pyw> und die Grafiken <v01.png> bis <v13.png>. Die entpackten Dateien müssen sich unter Android in demselben Ordner befinden bzw. in dem gleichen Pfad installiert bzw. kopiert werden, in dem sich auch das Hauptprogramm <Vermessung1.pyw> befindet, dies am Besten im Hauptspeicher. Sollte das Programm in Android nicht direkt geöffnet werden können, so ist über die Pydroid 3 IDE oben rechts zu starten mit:

> Open > Interal storage > <folder>[2] > <Vermessung1.pyw>

In Windows kann das Quellen-Programm auch in der Python <IDLE Shell 3.11.0> aufgerufen werden mit:

*Open ... > Dateipfad und Datei* (mit der Endung *.pyw)

mit der Befehlszeile: android = False

Da dies eine frei zugängliche Codierung ist, ist dringend davon abzuraten, sie ändern zu wollen. Ein reibungsloser Verlauf des Programms kann eingeschränkt oder verhindert werden. Eine Gewähr kann nicht übernommen werden.

Eine Druckausgabe ist nicht möglich. Eine dauerhafte Dokumentation dürfte mit einem <screenshot> möglich sein[3]. Allerdings sollten Screenshots, und damit eine dauerhafte Dokumentation, nur dann gemacht werden, wenn sich ein grüner Haken im Display neben der Programmüberschrift zeigt **und keine** Fehlermeldung angezeigt wird; nur um fehlerhafte Dokumentationen zu vermeiden.

## Hinweise zur Programmbedienung

Dezimalzahlen müssen durch einen Punkt und nicht mit einem Komma eingegeben werden (pythonspezifisch)[4]. Die Eingabe von Buchstaben ist nicht erlaubt. Durch eine programmspezifische Ausnahmebehandlung würde eine Berechnung nur durchgeführt werden, wenn eine Umwandlung der Eingabe in Gleitkommazahlen (ohne Buchstaben, Komma oder sonstige Zeichen) möglich wäre. Ansonsten müssen Veränderungen in der Eingabe erfolgen.

Vielfach ist in den Eingabefeldern ein Defaultwert (z. B. 100) angezeigt, der durch eine neue Eingabe überschrieben werden kann.

[2]directory, Ordner mit Programmen und Abbildungen (normalerweise im Hauptspeicher)

[3]Beispiele: für Windows <microsoft swift>; für Android <Screenshot touch> aus PlayStore

[4]Eine Einstellung des Punktes im Ziffernblock der Tastatur für Windows ist sinnvoll. Man wähle dazu in Windows: >Einstellungen > Zeit und Sprache > Sprache > Sprache hinzufügen > Deutsch-Optionen > Deutsch(Punkt). Hinweise im Internet, einen Punkt im Ziffernblock der Tastatur mit Tastaturtreiber anderer Länder (z.B. Schweiz) erzeugen zu wollen, wechseln nach Tastendruck zwar das Komma in einen Punkt, haben aber die Eigenschaft andere Sonderzeichen der Tastatur zu verändern. Deshalb kann es sinnvoll sein, softwaremäßig ausschließlich nur das Komma im Ziffernblock zu ändern: Die herunter ladbare Software PatchKeyboard_6.1 vermag das und erzeugt einen Tastaturtreiber <Deutsch(Punkt)>, der zu wählen wäre.

Durch den Button kann zur besseren Übersicht und zum Verständnis der Eingabefelder eine wieder schließbare Zeichnung im Display geöffnet werden. Grafiken sollten durch den unteren, sich rot färbenden Button geschlossen werden, weil sie sonst in der entsprechenden Instanz nicht erneut geöffnet werden können.

Ein gerade durchgeführter Berechnungsablauf wird neben der Titelüberschrift durch einen grünen Haken ✓ angezeigt. Ein Warnzeichen ⚠ taucht bei einer nachträglichen Betätigung einer Taste auf. Dann ist ein erneuter Berechnungsablauf **stets erforderlich**, da sonst im Zusammenhang fehlerhafte, nicht zusammen gehörige Angaben im Display erscheinen können.

Die Ausgabe erfolgt in [0,1 mgon] bzw. [mm] , wobei rechtsseitige Nullen nicht angezeigt werden.

Gelbe oder blassgelbe Eingabefelder dienen **ausschließlich einer Anzeige**. Eingaben darin haben keinerlei Einfluss auf die Auswertung.

Die einzelnen Klassen (class xxx()) sind modular und autark aufgebaut. Sie können eigenständig aus dem Verbund (Vermessung1.pyw) kopiert werden und in das folgende Skriptgerüst für Android eingefügt werden:

```
from tkinter import *
import math, decimal, tkinter.messagebox, os, sys
global android
android = True
class = xxx ()
    Inhalt der Klasse xxx (z. B. wie Inhalt von class StreckeUTM() ≙ <11 Strecke UTM ⇒ örtlich>
i = xxx () (Erzeugung einer Instanz)
```

## Vermessungstechnische Hinweise

Ein Maßstabsfaktor kann Eigenschaften einer Korrektion wie auch einer Reduktion haben: Ein Korrektionsfaktor berichtigt die ursprüngliche Messung in Annäherung an den wahren Wert (z. B. Kalibrierungswerte), ein Reduktionsfaktor behandelt z. B. die Horizontierung einer Strecke oder eine Abbildungsverzerrung (Verebnung).

Dem hier vielfach in der Eingabe vorkommenden Reduktionsfaktor $r$ (oder auch Maßstabsfaktor $m$) bei Strecken bzw. Koordinaten liegt folgende Definition zu Grunde:

$$reale\,\text{örtliche},\ horizontierte\ Strecke\,(\text{hor.})\ \times\ r\ =\ Strecke\,(\text{UTM})$$

# 1 Orthogonale und polare Elemente

Das Programm dient ebenso zur Überprüfung von Geradlinigkeiten, zur Ermittlung von Abständen und für einen Senkrechtschnitt. Es können auch ausschließlich nur Richtungswinkel und Entfernung berechnet werden.

Gegeben:

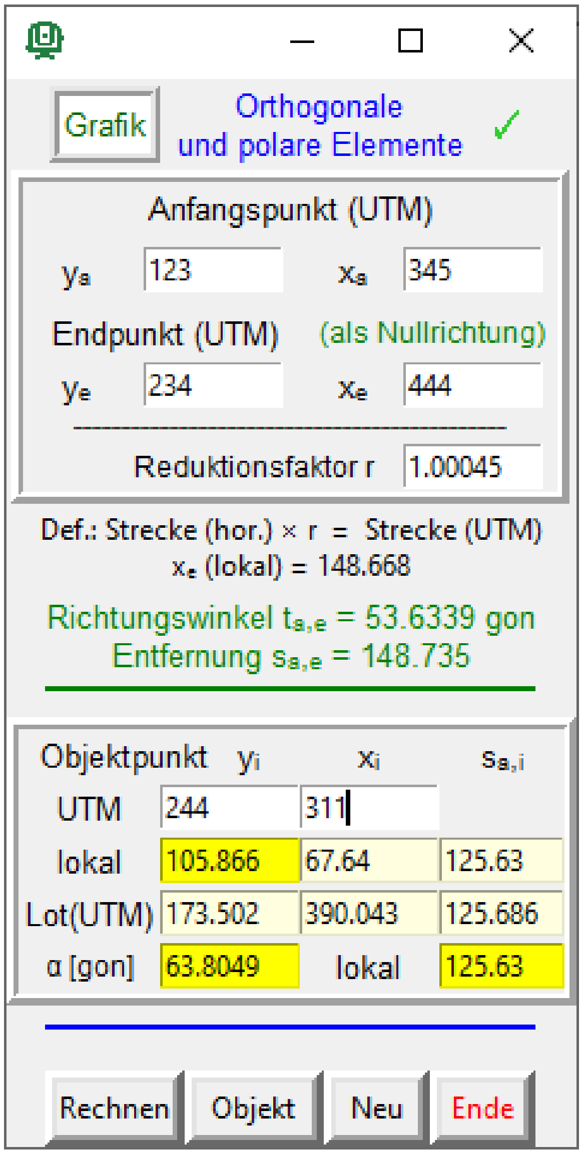
$y_{\mathrm{a}}, x_{\mathrm{a}}$ Anfangspunkt (i.d.R. UTM)
$y_{\mathrm{e}}, x_{\mathrm{e}}$ Endpunkt (UTM)
$y_i, x_i$ Objektpunkt (UTM)
$r$ Reduktionsfaktor (Default = 1)

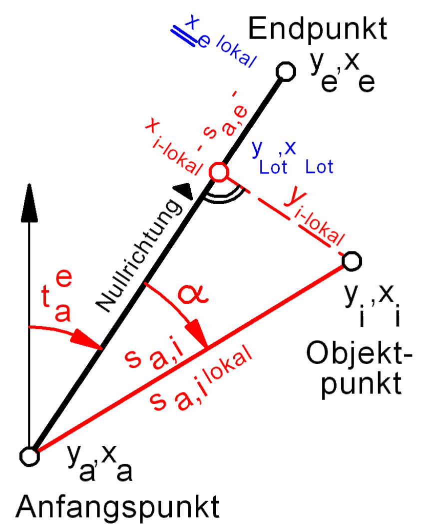
Gesucht:

$t_{\mathrm{a}}^{\mathrm{e}}, s_{\mathrm{a}}^{\mathrm{e}},$ Richtungswinkel [gon] und Entfernung
$y_{i-\mathrm{lokal}}, x_{i-\mathrm{lokal}}$ lokale orthogonale Elemente
($y_{i-\mathrm{lokal}}$ zur Überprüfung der Geradlinigkeit)
(point to line)
$\alpha, s_{\mathrm{a},i}$ bzw. $s_{\mathrm{a},i\,\mathrm{lokal}}$ Polarwinkel, Strecken
(Polarelemente)

Zusätzliche Ergebnisse:

$y_{i-\mathrm{Lot}}, x_{i-\mathrm{Lot}}$ Koordinaten des Lotfußpunktes (UTM)
$x_{\mathrm{e}}$ lokales Endmaß



Die Nullrichtung für die Polarelemente $\alpha, s_{\mathrm{a},i\,\mathrm{lokal}}$ ist von $y_\mathrm{a}, x_\mathrm{a}$ nach $y_\mathrm{e}, x_\mathrm{e}$ gegeben.

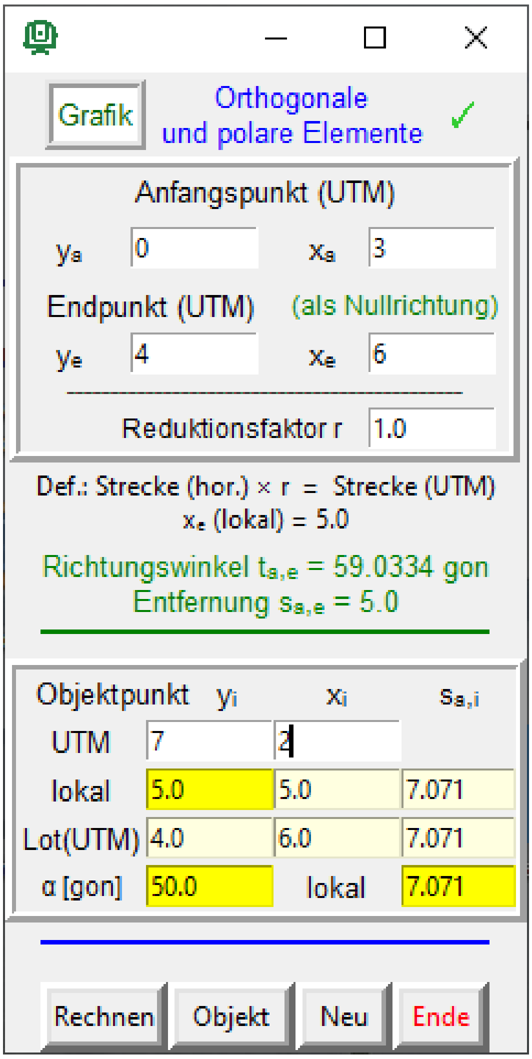
**Achtung:**

> Orthogonale und polare Elemente werden i.d.R. für örtliche Absteckungen benötigt. Werden als Ausgangswerte für $y_\mathrm{a}, x_\mathrm{a}, y_\mathrm{e}, x_\mathrm{e}$ UTM-Koordinaten benutzt, so bewirkt die Eingabe des Reduktionsfaktors r laut der Definition
>
> $$Strecke(\mathrm{hor.}) \times r = Strecke(\mathrm{UTM})$$
>
> hier eine korrekte Berechnung[5], also
>
> $$Elemente(\mathrm{horizontal,lokal}) \Longleftrightarrow Koordinaten(\mathrm{UTM}) : r$$
>
> Sollte der wohl seltene, umgekehrte Weg beschritten werden, ist als Reduktionsfaktor dessen reziproker Wert (also $1/r$) zu nehmen.

Bei einem Reduktionsfaktor $r = 1$ liegen eingegebene Koordinaten und die Rechenergebnisse auf einer Rechenebene (s. rechte Abbildung).

Für die alleinige Berechnung von Richtungswinkel und Entfernung reichen die Angaben $y_\mathrm{a}, x_\mathrm{a}, y_\mathrm{e}, x_\mathrm{e}$. Desgleichen können Richtungswinkel und Entfernung können auch mit dem Programm 3 <Polaraufnahme (Koordinaten)> berechnet werden.

Die Berechnung von Senkrechtschnitten (über den Lotfußpunkt) sind auch mit diesem Programmteil möglich.

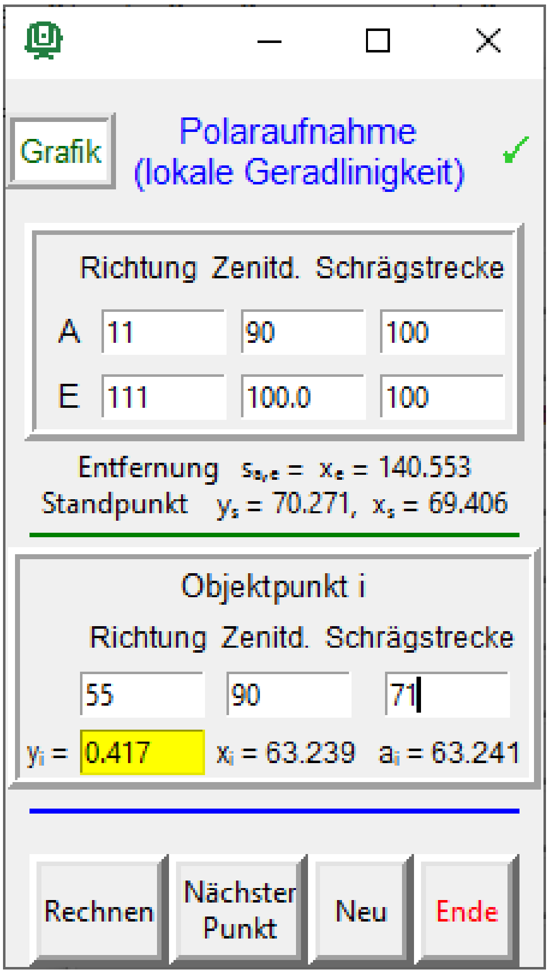



[5] d. h. $s_{\mathrm{a,e}}$ wird im übergeordneten System (i.d.R. UTM) gerechnet, dagegen $x_\mathrm{e}(\mathrm{lokal})$ im örtlichen, unter Berücksichtigung des Reduktionsfaktors $r$. Die Werte für $s_{\mathrm{a,e}}$ und $x_\mathrm{e}(\mathrm{lokal})$ wie auch $s_{\mathrm{a},i}$ und $s_{\mathrm{a},i(\mathrm{lokal})}$ müssen deshalb nicht identisch sein.

## 2 Polaraufnahme (lokale Geradlinigkeit)

Gegeben:

lokale Richtungen und Zenitdistanzen [gon] und Schrägstrecken zum
$r_a, Zenitdistanz_a, s_a$ Anfangspunkt
$r_e, Zenitdistanz_e, s_e$ Endpunkt
$r_i, Zenitdistanz_i, s_i$ Objektpunkt
im örtlichen System
Defaultwerte für Zenitdistanzen = 100 [gon]

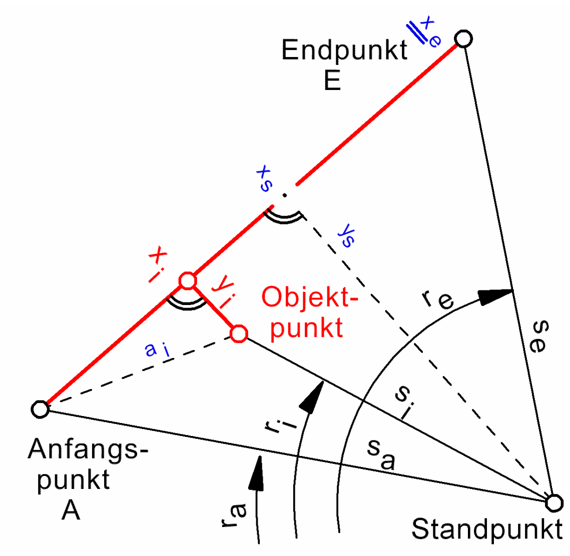



Gesucht:

$y_i, x_i$ (lokale orthogonale Elemente)
($y_i$ zur Überprüfung der Geradlinigkeit und der Absteckung von Geraden und Parallelen)

Zusätzliche Ergebnisse:

$x_e$ (lokal) Endmaß
$y_s, x_s$ (lokal) Standpunktkoordinaten
$a_i$ (lokal) Strecke



## 3 Polaraufnahme (Koordinaten)

Gegeben:

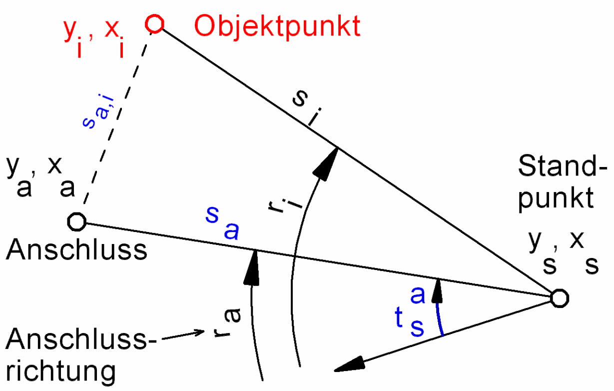
$y_s, x_s$ Standpunkt
$y_a, x_a, r_a$ Anschlusspunkt mit Richtung [gon]
$m$ Maßstabsfaktor (Default = 1)
$r_i, Zenitdistanz_i, s_i$ Objektpunkt
mit Richtung und Zenitdistanz [gon]
(Default = 100 [gon]) und Schrägstrecke

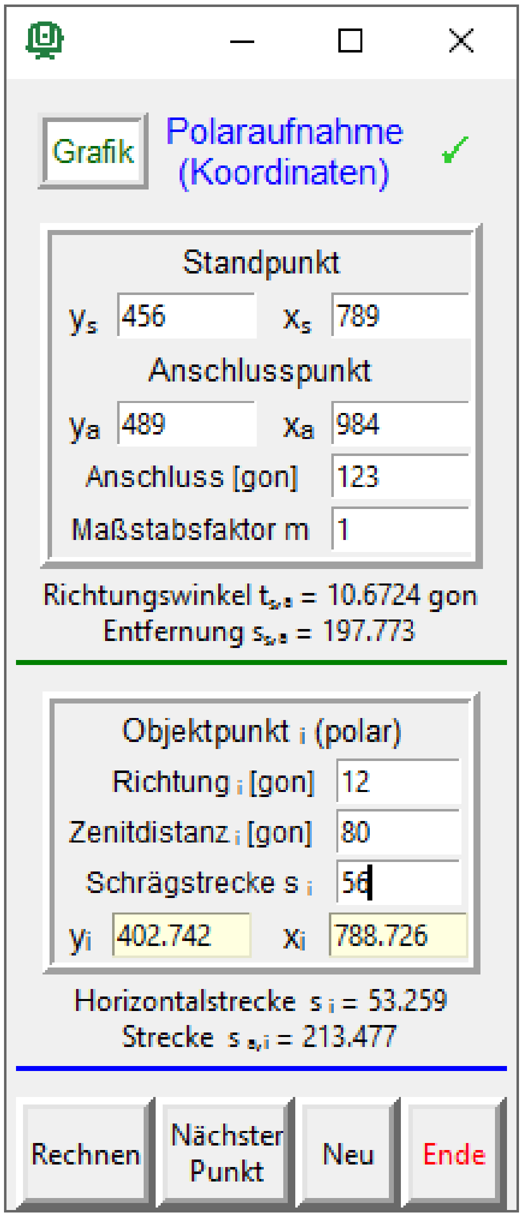
Gesucht:

$y_i, x_i$ Objektpunkt

Zusätzliche Ergebnisse:

$t_s^a$ Anschlussrichtungswinkel [gon]
(aus Koordinaten)
$s_{s,a}$ Strecke Stand- zum Anschlusspunkt
(Entfernung)
$s_{a,i}$ Strecke Anschluss- zum Objektpunkt




Soll eine örtliche Aufnahme in UTM-Koordinaten berechnet werden, ist zunächst der Reduktions-(Maßstabs-)faktor zu ermitteln und einzutragen.

## 4 Transformation, Kleinpunkte

Gegeben:

$\mathrm{A}y_\mathrm{a}, \mathrm{A}x_\mathrm{a}$ Anfangspunkt System A
$\mathrm{A}y_\mathrm{e}, \mathrm{A}x_\mathrm{e}$ Endpunkt System A
$\mathrm{A}y_i, \mathrm{A}xi_i$ Objektpunkt

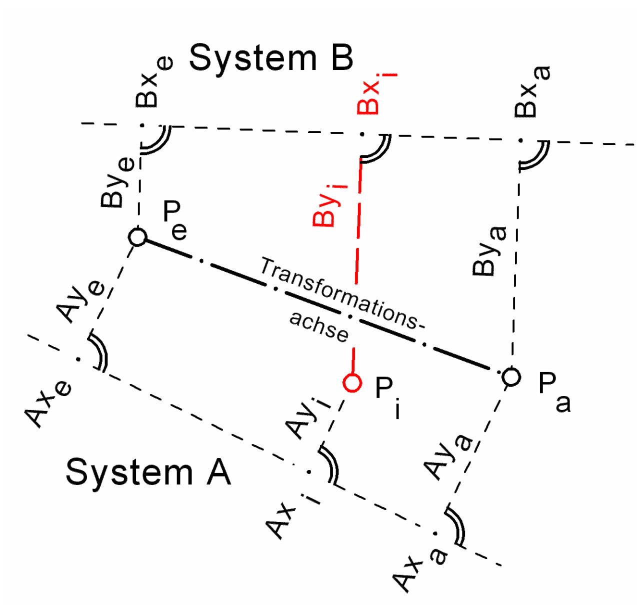
$\mathrm{B}y_\mathrm{a}, \mathrm{B}x_\mathrm{a}$ Anfangspunkt System B
$\mathrm{B}y_\mathrm{e}, \mathrm{B}x_\mathrm{e}$ Endpunkt System B

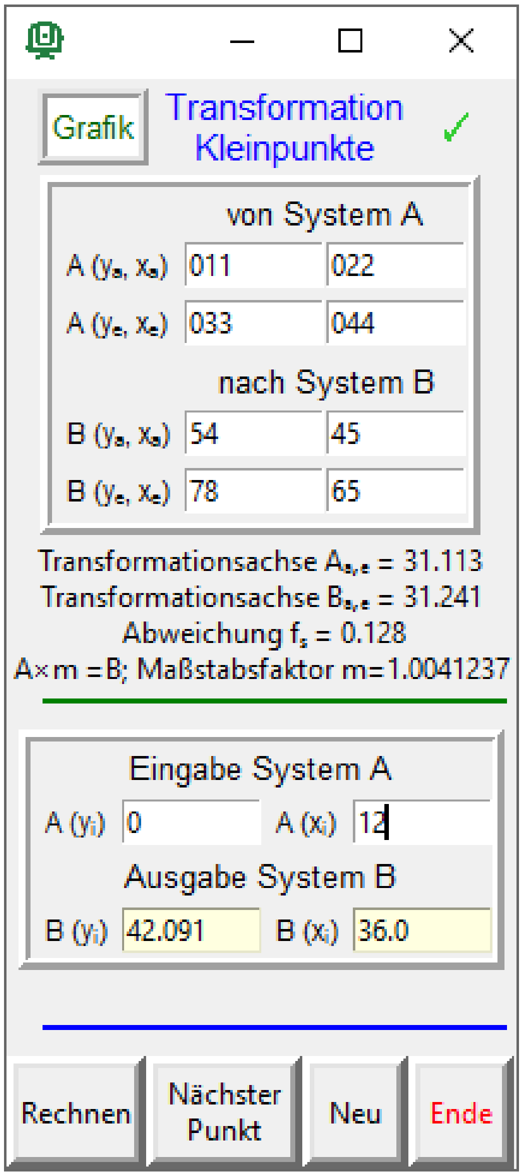
Gesucht:

$\mathrm{B}y_i, \mathrm{B}x_i$ Objektpunkt im System B

Zusätzliche Ergebnisse:

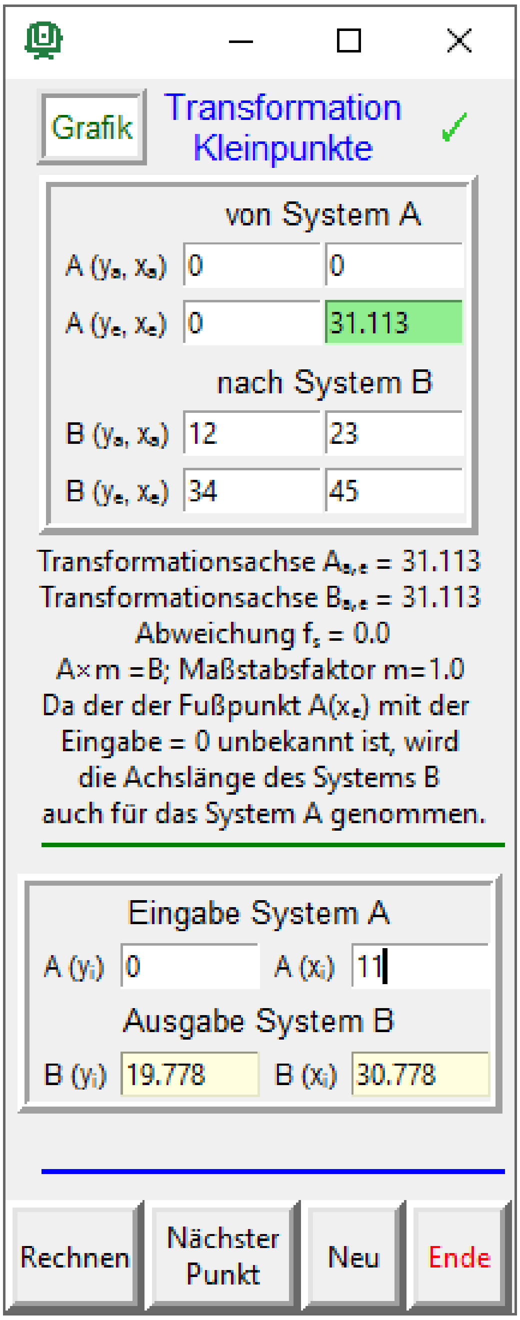
Längen der Transformationsachse
im System A und B $\rightarrow \mathrm{A}s_{\mathrm{a,e}}, \mathrm{B}s_{\mathrm{a,e}}$
$f_s$ = Abweichung $(\mathrm{B}s_{\mathrm{a,e}} - \mathrm{A}s_{\mathrm{a,e}})$
$m$ Maßstabsfaktor




Hiermit können lineare Achstransformationen wie auch übliche Kleinpunktberechnungen durchgeführt werden; Kleinpunktberechnungen derart, dass in den entsprechenden Eingabefeldern Nullen eingetragen werden, die hier i. d. R. als Defaultwerte zur Verfügung stehen.

Nebenstehend ist ein Beispiel einer Kleinpunktberechnung minimalster Form, mit nicht bekanntem Endmaß $\mathrm{A}x_\mathrm{e}$ im System A (Eingabe $\mathrm{A}x_\mathrm{e} = 0$) und mit vielen Nullen als Defaultwerte.



## 5 Geradenschnitte

Gegeben:

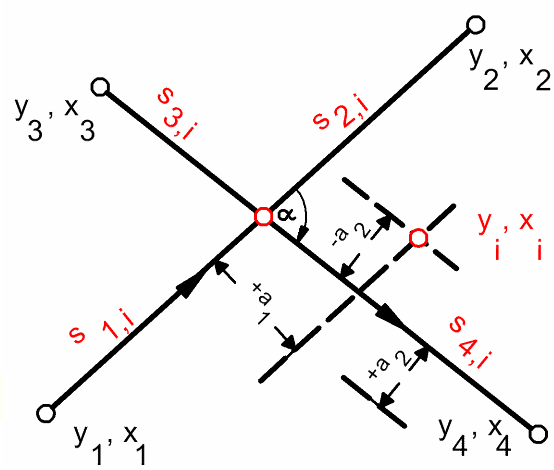
$(y_1, x_1) \rightarrow (y_2, x_2)$ Gerade 1
$\pm a_1$ Abstand
$(y_3, x_3) \rightarrow (y_4, x_4)$ Gerade 2
$\pm a_2$ Abstand

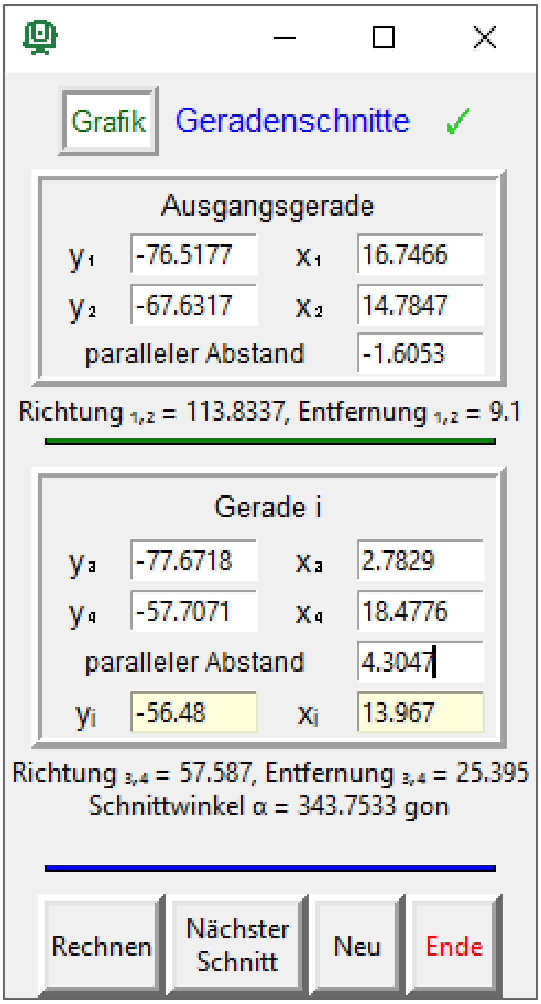
Gesucht:

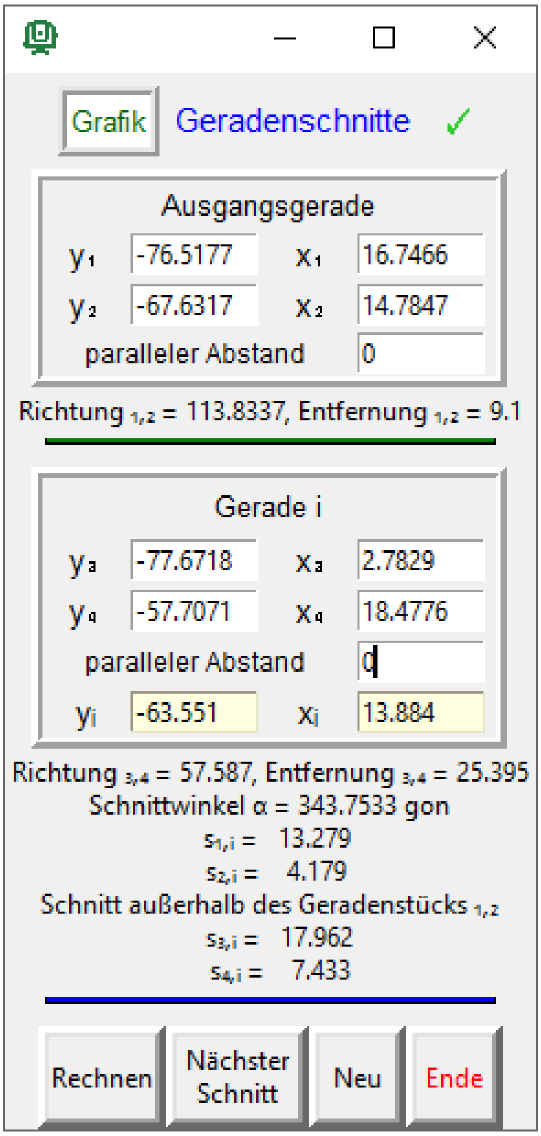
$y_s, y_s$ Schnittpunkt

Zusätzliche Ergebnisse:

$\alpha$ Schnittwinkel [gon]
$s_{1,i}, s_{2,i}, s_{3,i}, s_{4,i}$ Streckenanteile, sofern keine Abstände gegeben sind



## 6 Kreis-Geraden-Schnitt

Gegeben:

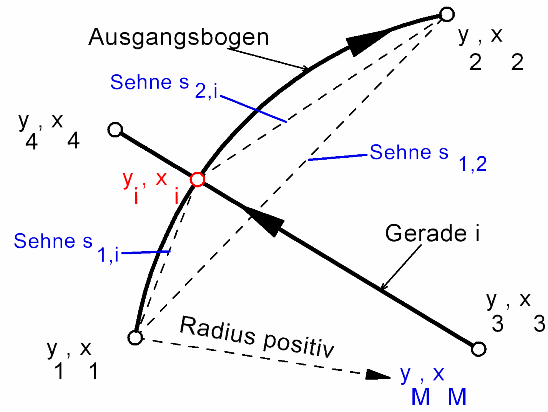
$y_1, x_1$ Anfang des Bogens (rechtsläufig)
$y_2, x_2$ Ende des Bogens
$Radius$ +
$(y_3, x_3) \rightarrow (y_4, x_4)$ Gerade

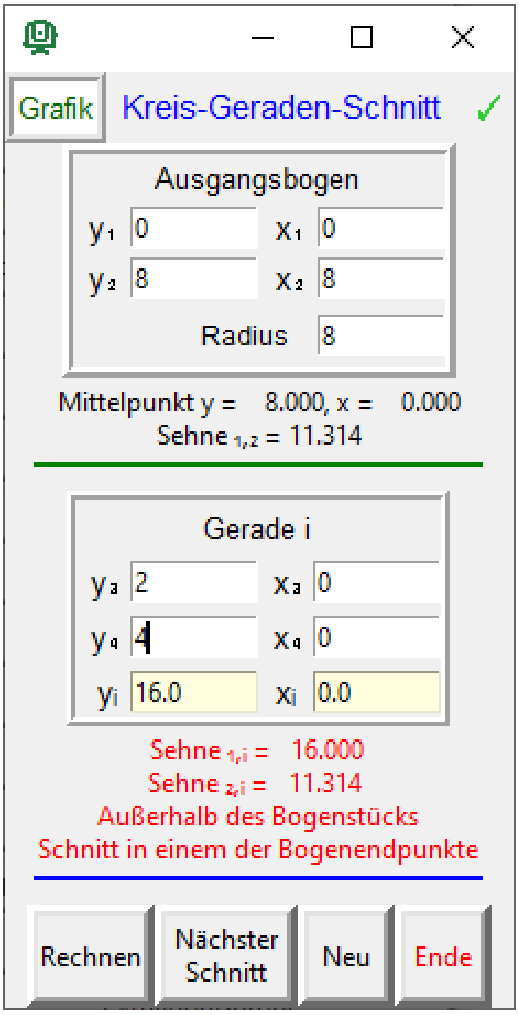
Gesucht:

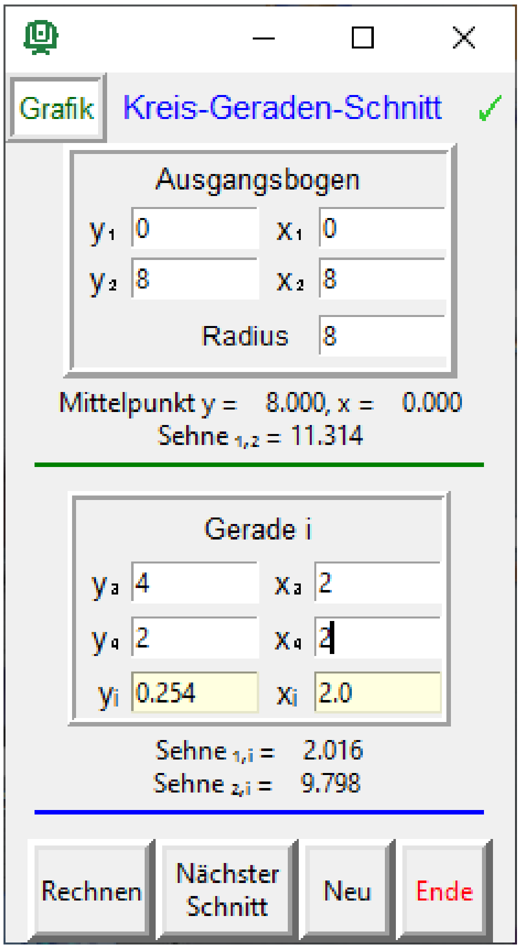
$y_i, y_i$ Schnittpunkt

Zusätzliche Ergebnisse:

$s_{1,i}, s_{2,i}, s_{1,2}$ Sehnenlängen
$y, x$ Mittelpunktkoordinaten




Werden beide Schnittpunkte benötigt, sind im nächsten Durchlauf $(y_3, x_3)$ und $(y_4, x_4)$ zu vertauschen.

## 7 Kreisbogenteilung

Gegeben[6]:

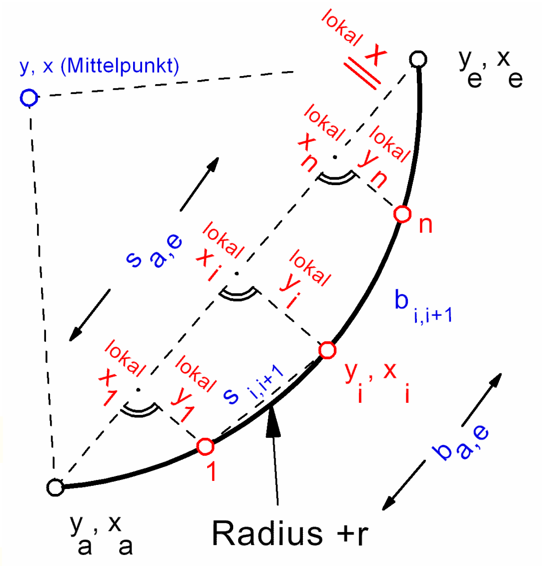
$y_\mathrm{a}, x_\mathrm{a}$ Anfang des Bogens (linksläufig)
$y_\mathrm{e}, x_\mathrm{e}$ Ende des Bogens
oder $s_\mathrm{a,e}$
$m$ Maßstabsfaktor (Default = 1)
$Radius$ +
$n$ Anzahl der Zwischenpunkte (für $n < 10$)

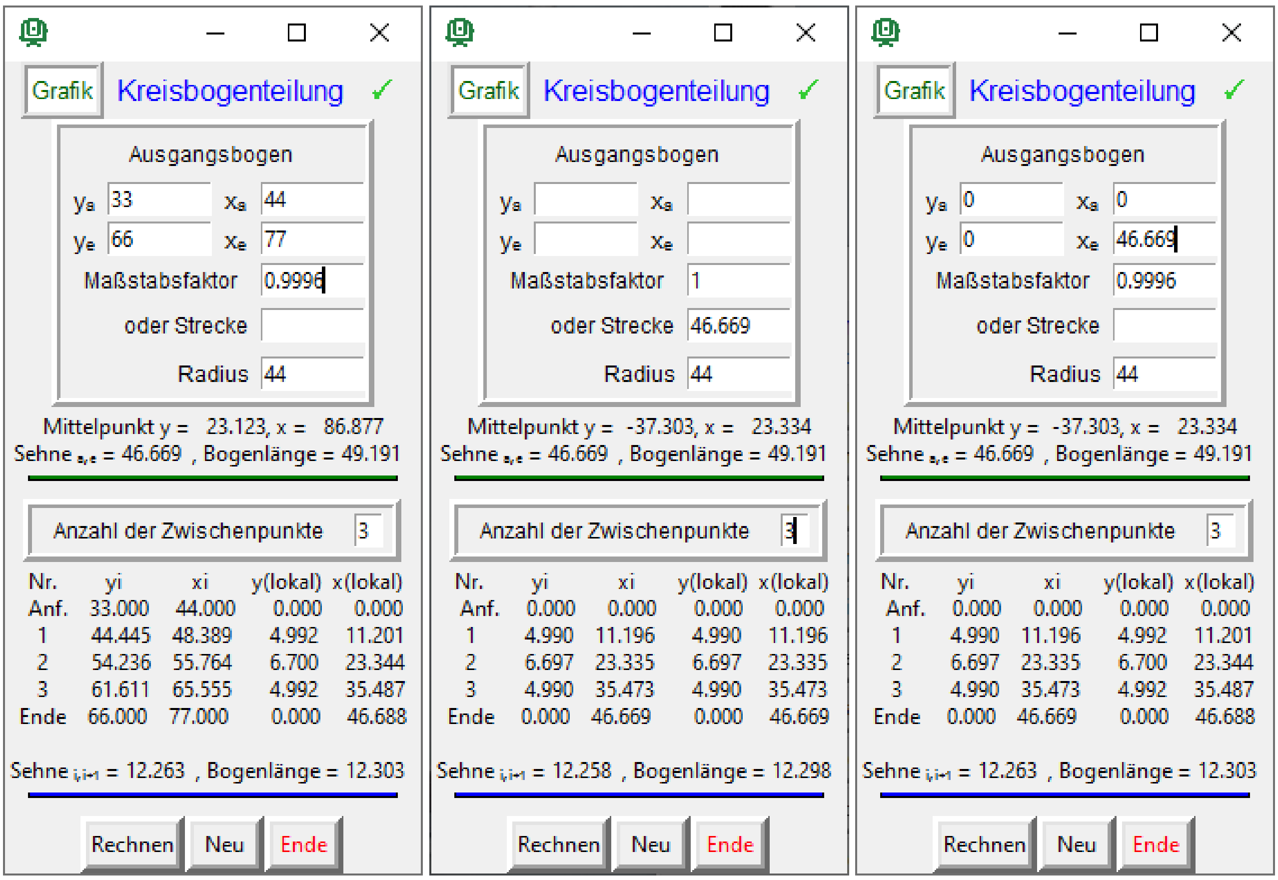
Gesucht:

$y_i, x_i$ Objektpunkt

Zusätzliche Ergebnisse:

$s_\mathrm{a,e}, s_{i,i+1}$ Sehnenlängen
$b_\mathrm{a,e}, b_{i,i+1}$ Bogenlängen
$y, x$ Mittelpunktkoordinaten




a) b) c)

[6]Damit die lokalen Ordinaten positiv werden, wurde der Bogen linksläufig eingeführt.

Sollen örtliche Absteckungselemente aus UTM-Koordinaten berechnet werden, ist zunächst der Reduktions-(Maßstabs-)faktor zu ermitteln und neben den Koordinaten des Ausgangsbogens einzugeben [Beispiel a)]; lt. der Definition (s. Seite 4 und 18):

$$Elemente(\text{horizontal, lokal}) \Longleftrightarrow Koordinaten(\text{UTM}) : r \text{ (bzw. } m).$$

Dagegen wird die Eingabe einer Ausgangsstrecke **unbedingt vorrangig** gegenüber den Eingabekoordinaten berücksichtigt und erzeugt automatisch einen Maßstabsfaktor 1. $y, x$ sind dann örtliche, lokale Koordinaten des Mittelpunktes [Beispiel b)]. Liegt eine UTM-Strecke als Sehne vor und sollen örtliche, lokale Elemente berechnet werden, sind neben dem Reduktionsfaktor folgende Werte einzugeben: $y_a, x_a, y_e = 0$ und die Strecke als $x_e$ [Beispiel c)].

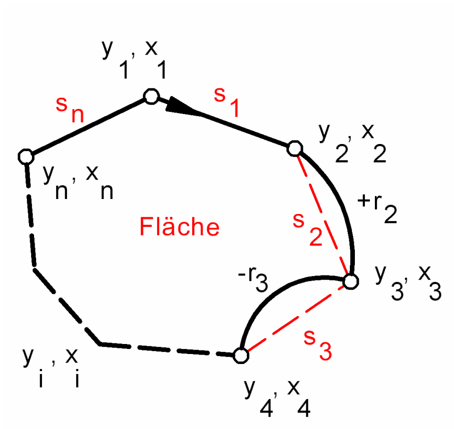
## 8 Flächenberechnung

Gegeben:

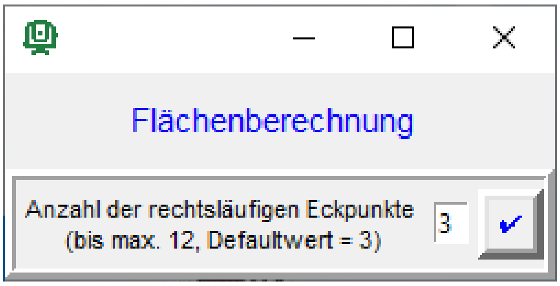
$y_i, x_i$ rechtsläufige $n$ Eckpunkte
$+Radius \rightarrow$ Das Kreissegment wird addiert.
$-Radius \rightarrow$ Das Kreissegment wird subtrahiert.
$n$ Anzahl der rechtsläufigen Eckpunkte
$(3 \leq n \leq 12)$

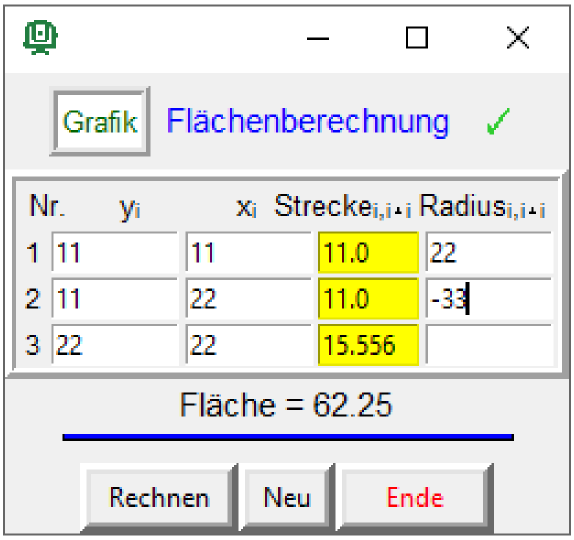
Gesucht:

$Fläche$
$s_i$ Seiten- bzw. Sehnenlängen



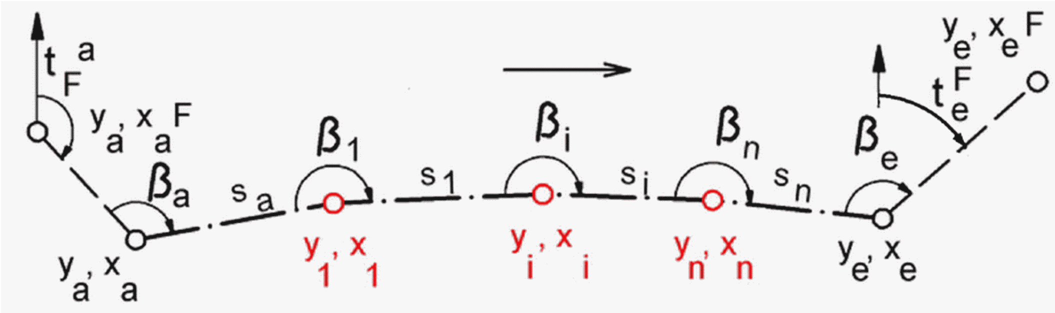
n = Anzahl der Punkte



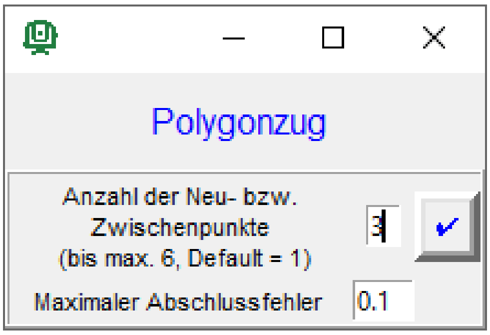

Bei eine aus örtlichen Koordinaten ermittelten Fläche auf eine beispielsweise UTM-Bezugsfläche zu transformieren, ist diese mit dem Ouadrat des Reduktionsparameters zu multiplizieren.

# 9 Polygonzug



Gegeben (je nach Variante):

$y_{\mathrm{a}}^{\mathrm{F}}, x_{\mathrm{a}}^{\mathrm{F}}$ Fernziel (Koordinaten)
oder $t_{\mathrm{F}}^{\mathrm{a}}$ Anschlussrichtungswinkel
$y_{\mathrm{a}}, x_{\mathrm{a}}$ Anfangspunkt
$y_{\mathrm{e}}, x_{\mathrm{e}}, y_{\mathrm{e}}^{\mathrm{F}}, x_{\mathrm{e}}^{\mathrm{F}}$ Endpunkt, Fernziel
oder $t_{\mathrm{e}}^{\mathrm{F}}$ Abschlussrichtungswinkel
$\beta_{\mathrm{a}}, \beta_i, \beta_{\mathrm{e}}$ Brechungswinkel [gon]
$n$ Anzahl der Neu- bzw. Zwischenpunkte
$(1 \leq n \leq 6)$
$maximaler Abschlussfehler$
$s_{\mathrm{a}}, \mathrm{s}_i$ Polygonseiten $(i = 1$ nach $n)$
Maßstabsfaktor (Default = 1) **vor** der Transformation




Gesucht:

$y_i, x_i$

Zusätzliche Ergebnisse:

Abweichungen des Koordinatenabschlusses $f_y, f_x$
Winkelschlussabweichung $f\beta$, Längsabweichung $L$,
Querabweichung $W$
Maßstabsfaktor nach der Transformation

Das Programm erfordert zwei Durchläufe. Beim ersten Durchlauf werden vorläufige Koordinaten berechnet, auch zur Anzeige von Längs-, Quer- und Winkelschlussabweichung [Beispiel a)] je nach Eingabevariante.

Endgültige Koordinaten werden im zweiten Durchlauf [Beispiel b)] durch eine Transformation (Drehung und Streckung) erzeugt, womit eine Verteilung der Abweichungen erfolgt. Werden nach dieser Transformation $\mathrm{f}_y = 0, \mathrm{f}_x = 0, L = 0, W = 0$ angezeigt, ist die Transformation korrekt verlaufen. Der zweite, grüne Haken neben der Titelüberschrift <Polygonzug> kennzeichnet die Durchführung des zweiten Programmdurchlaufs. Die anfängliche Eingabe eines maximalen Abschlussfehlers ist aus Sicherheitsgründen erforderlich [Beispiel c)].

Die Eingabefelder dienen teilweise gleichzeitig zur Programmsteuerung und bieten Berechnungs**varianten**:

- Der Anschluss kann wahlweise über den Anschlussrichtungswinkel $t_{\mathrm{F}}^{\mathrm{a}}$ oder über die Koordinaten von $y_{\mathrm{a}}^{\mathrm{F}}, x_{\mathrm{a}}^{\mathrm{F}}$ erfolgen. Treten beide Angaben auf, wird der Anschlussrichtungswinkel $t_{\mathrm{F}}^{\mathrm{a}}$ gegenüber den Koordinaten von $y_{\mathrm{a}}^{\mathrm{F}}, x_{\mathrm{a}}^{\mathrm{F}}$ vorrangig bearbeitet. Das gilt analog auch für den Abschluss bezüglich $y_{\mathrm{e}}^{\mathrm{F}}, x_{\mathrm{e}}^{\mathrm{F}}$ bzw. $t_{\mathrm{e}}^{\mathrm{F}}$.

- $\beta_{\mathrm{a}} = 0; \Longrightarrow$ ohne Winkelanschluss (Eingaben $y_{\mathrm{a}}^{\mathrm{F}}, x_{\mathrm{a}}^{\mathrm{F}}$ bzw. $t_{\mathrm{F}}^{\mathrm{a}}$ sind unnötig)

  Wegen der fehlenden Anschlussrichtung wird hier programmintern eine fingierte in Richtung des Endpunktes genommen. Die vor der Transformation ermittelte Werte für $f_y, f_x, L, W$ sind deshalb **irregulär**. Ringpolygone können nicht gerechnet werden.

- $\beta_e = 0; \Longrightarrow$ ohne Winkelabschluss (Eingaben $y_{\mathrm{e}}\mathrm{F}, x_{\mathrm{e}}\mathrm{F}$ sind unnötig)

- $y_{\mathrm{e}}$ und $x_{\mathrm{e}} = 0; \Longrightarrow$ ohne Koordinatenabschluss (Eingaben von $\beta_{\mathrm{e}}, y_{\mathrm{e}}^{\mathrm{F}}, x_{\mathrm{e}}^{\mathrm{F}}$ sind unnötig)
  $y_{\mathrm{e}}$ und $x_{\mathrm{e}}$ werden berechnet.

- $y_{\mathrm{a}} = y_{\mathrm{e}}$ und $x_{\mathrm{a}} = x_{\mathrm{e}}; \Longrightarrow$ Ringpolygon

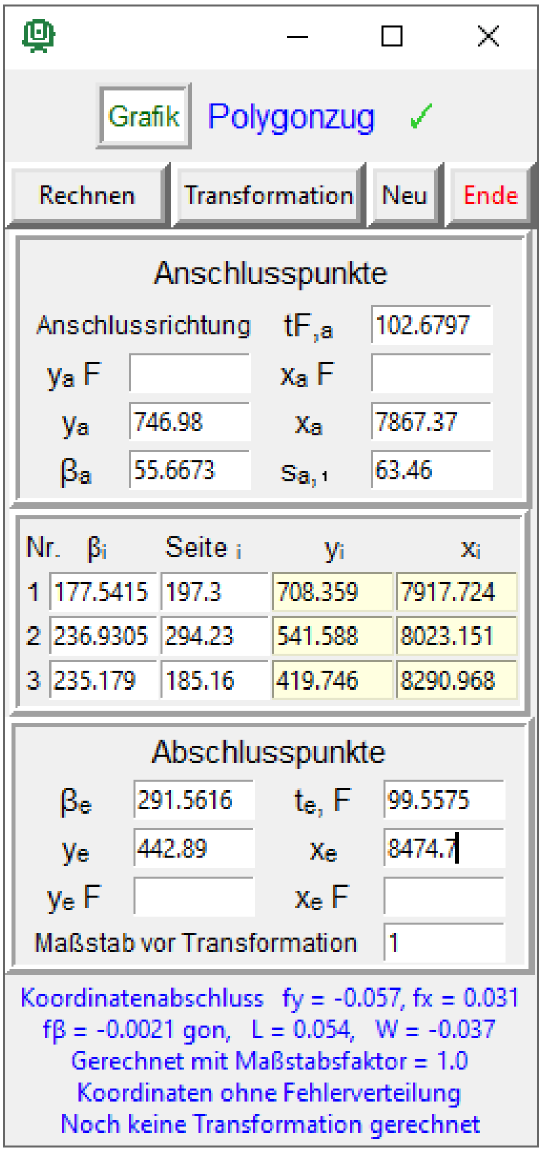
  (Anschlüsse mit $y_{\mathrm{a}}^{\mathrm{F}}, x_{\mathrm{a}}^{\mathrm{F}}$ oder $t_{\mathrm{F}}^{\mathrm{a}}$ und $\beta_{\mathrm{a}} \neq 0$ müssen gegeben sein.)

  $\Longrightarrow$ Ringpolygone sollten möglichst, wegen der unglücklichen Verteilung der Abweichungen und nicht machbarer Maßstabskontrolle vermieden werden. Eine Transformation zu deren Verteilung kann in diesem Fall nicht gerechnet werden. Bei der Betätigung der Taste <Transformation> erfolgt die Verteilung der Abweichungen dann proportional zur Länge der Polygonseiten: $\mathrm{d}(\Delta y_i) = \frac{s_i}{[s]} * fy$ und $\mathrm{d}(\Delta x_i) = \frac{s_i}{[s]} * fx$.

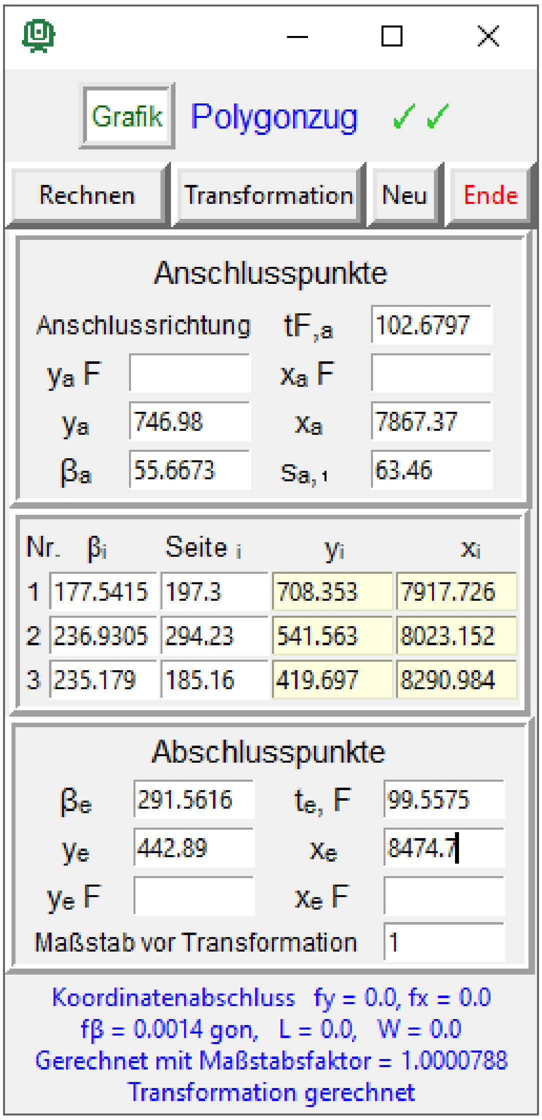
Eine einfache Fehleranalyse kann durchgeführt werden, wenn der Polygonzug nicht mit dem Defaultwert von 1 für den Maßstabsfaktor gerechnet wird, sondern zunächst als Maßstabsfaktor die zuvor ermittelte Maßstabsverzerrung (lokal nach UTM) genommen wird. Erst im zweiten Durchlauf erfolgt die endgültige Verteilung der Abweichungen dann durch die Transformation.

Vor (eingegeben) und nach (berechnet) der Transformation:

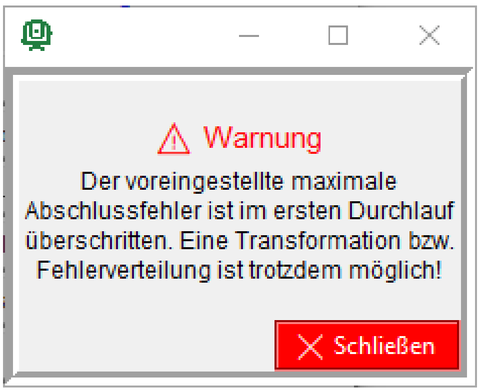
$f\beta =$ Winkelabschlussabweichung [gon] $\Longrightarrow t_{\mathrm{e,Soll}} - t_{\mathrm{e,Ist}}$
$f_y, f_x =$ Abweichungen des Koordinatenabschlusses
$L =$ Längsabweichung
$W =$ lineare Querabweichung
$f_s =$ lineare Abschlussabweichung $\Longrightarrow \sqrt{f_y^2 + f_x^2} = \sqrt{L^2 + W^2}$
(Vor der Transformation erfolgt keine Verteilung der Winkelabschlussabweichung;
eine Verteilung der Winkelabweichung kann zu einer Verbiegung des Zuges führen.)
Maßstabsfaktor, $\Longrightarrow L = (q - 1) * s_{\mathrm{a,e}}$


Grafik
Polygonzug ✓
Rechnen
Transformation
Neu
Ende
Anschlusspunkte
Anschlussrichtung tF,a 102.6797
ya F
xa F
ya 746.98
xa 7867.37
βa 55.6673
Sa,1 63.46
Nr. βi Seite i yi xi
1 177.5415 197.3 708.359 7917.724
2 236.9305 294.23 541.588 8023.151
3 235.179 185.16 419.746 8290.968
Abschlusspunkte
βe 291.5616
te, F 99.5575
ye 442.89
xe 8474.7
ye F
xe F
Maßstab vor Transformation 1
Koordinatenabschluss fy = -0.057, fx = 0.031
fβ = -0.0021 gon, L = 0.054, W = -0.037
Gerechnet mit Maßstabsfaktor = 1.0
Koordinaten ohne Fehlerverteilung
Noch keine Transformation gerechnet

a)

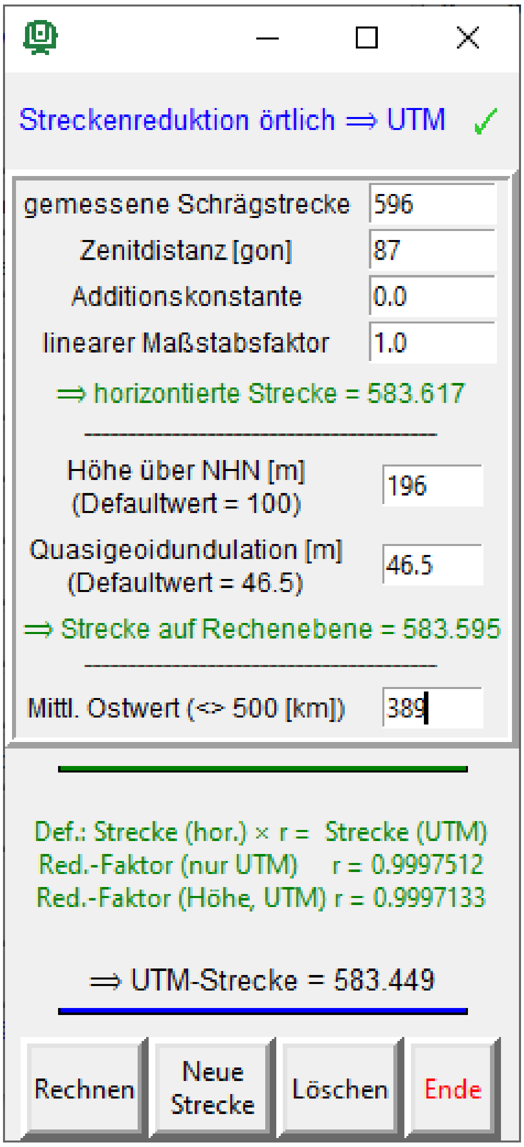

Grafik
Polygonzug ✓ ✓
Rechnen
Transformation
Neu
Ende
Anschlusspunkte
Anschlussrichtung tF,a 102.6797
ya F
xa F
ya 746.98
xa 7867.37
βa 55.6673
Sa,1 63.46
Nr. βi Seite i yi xi
1 177.5415 197.3 708.353 7917.726
2 236.9305 294.23 541.563 8023.152
3 235.179 185.16 419.697 8290.984
Abschlusspunkte
βe 291.5616
te, F 99.5575
ye 442.89
xe 8474.7
ye F
xe F
Maßstab vor Transformation 1
Koordinatenabschluss fy = 0.0, fx = 0.0
fβ = 0.0014 gon, L = 0.0, W = 0.0
Gerechnet mit Maßstabsfaktor = 1.0000788
Transformation gerechnet

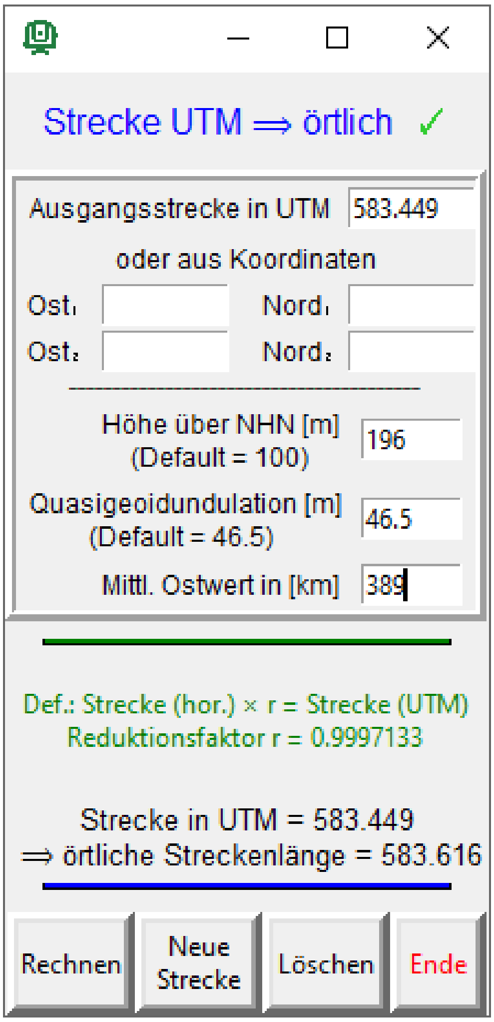
b)


Warnung
Der voreingestellte maximale Abschlussfehler ist im ersten Durchlauf überschritten. Eine Transformation bzw. Fehlerverteilung ist trotzdem möglich!
Schließen

c)

## 10 Streckenreduktion örtlich ⇒ UTM
## 11 Strecke UTM ⇒ örtlich



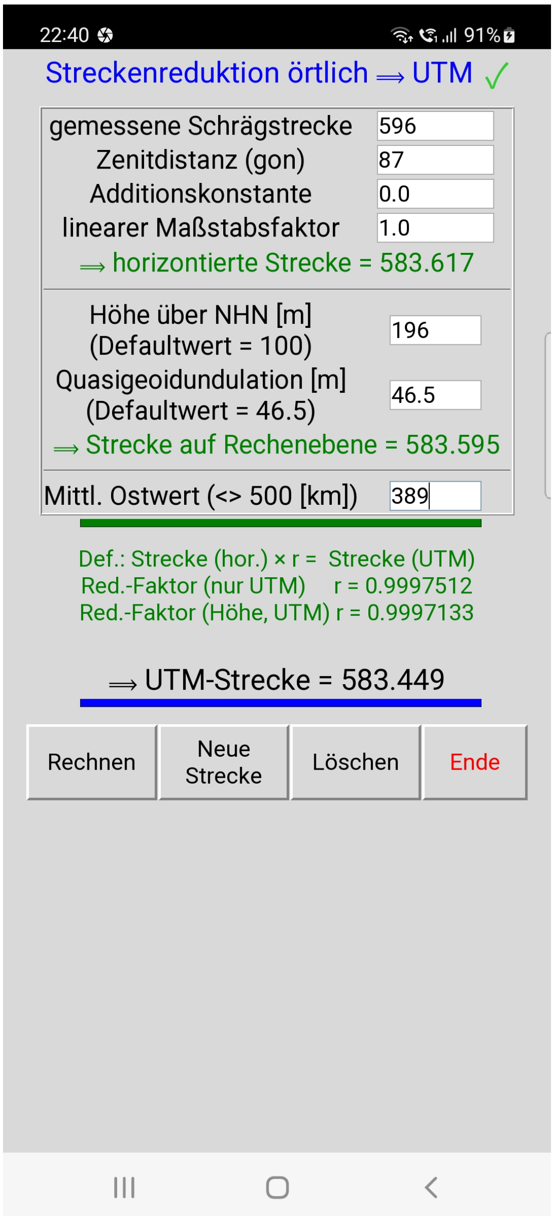



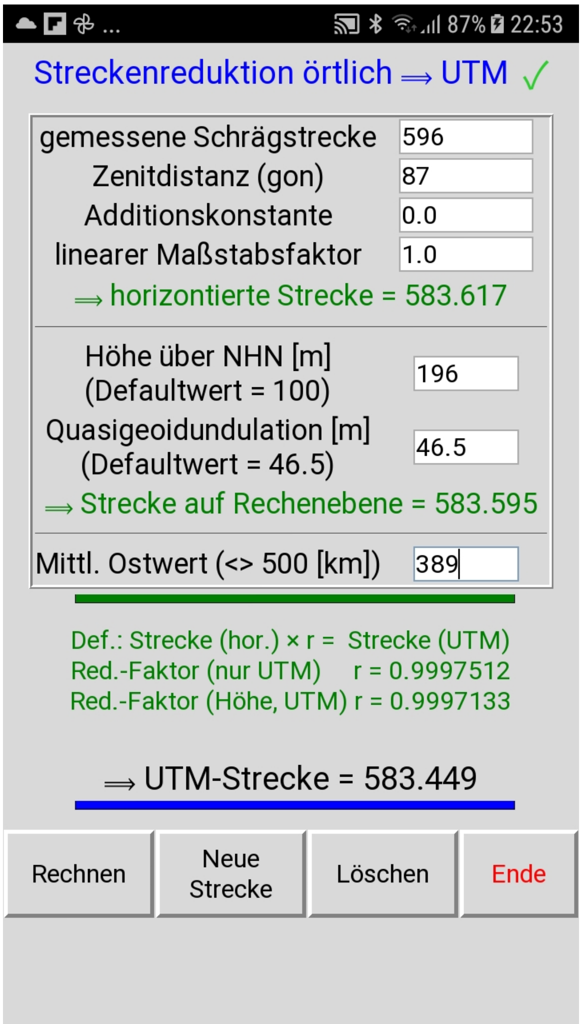

$$S_{[\text{örtlich}]} = \frac{\sqrt{\left((Ost_2 - Ost_1)^2 + (Nord_2 - Nord_1)^2\right)} \; = S_{[\text{UTM}]}}{\left(1 + \frac{\left(Ost_{\text{m[km]}} - 500\right)^2}{2R^2_{[\text{km}]}} - \frac{h_{\text{NHN[m]}} + h_{\text{Undulation[m]}}}{R_{[\text{km}]} * 1000}\right) * 0.9996 \; = r}$$

Eine nicht eingegebene Höhe über NHN erzeugt einen Defaultwert von 100. Eine fehlerhafte Höhenangabe von 100 m würde an der Streckenreduktion bei 100 m einen Fehler von 1,6 mm bewirken.

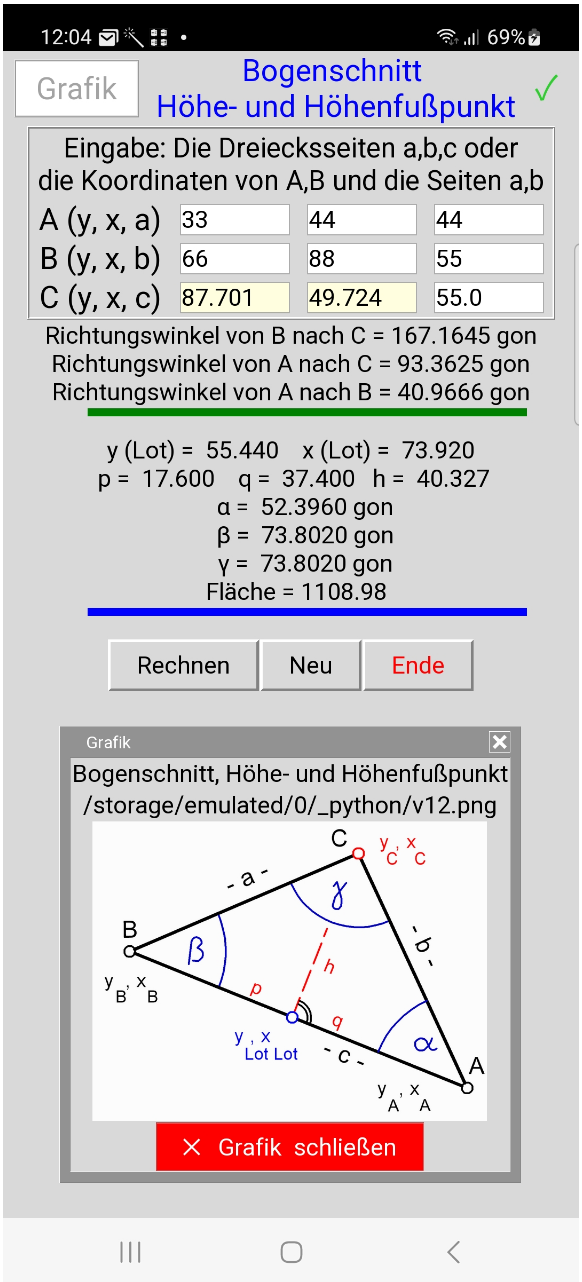
Für die Undulation wurde ein veränderbarer Defaultwert von 46.5 vorgegeben (Kölner Dom). Intern wurde ein mittlerer Erdradius von 6381 km verwendet (NRW).

Ein mittlerer Ostwert in [km] ist unbedingt einzugeben. Bei einem mittl. Ostwert von 319.5 bzw. 680.5 ist die Abbildungsverzerrung = 0.




Screenshot von Samsung Galaxy A52s 5G (links) und Samsung Galaxy J5 (rechts)

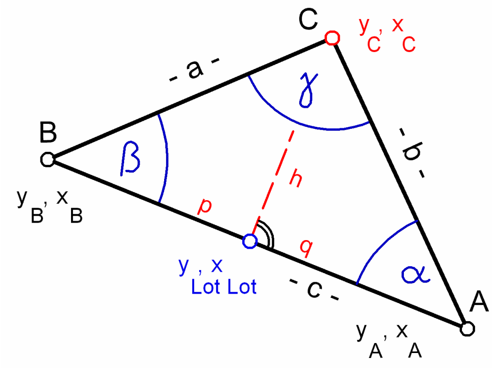

Screenshot von Samsung Galaxy Tab S4 (links) und von Samsung Galaxy A52s 5G (rechts)

# 12 Bogenschnitt

### Höhe und Höhenfußpunkt

Gegeben:

$y_A, x_A, y_B, x, a, b$ Koordinaten und Seiten
oder
$a, b, c$ Dreiecksseiten

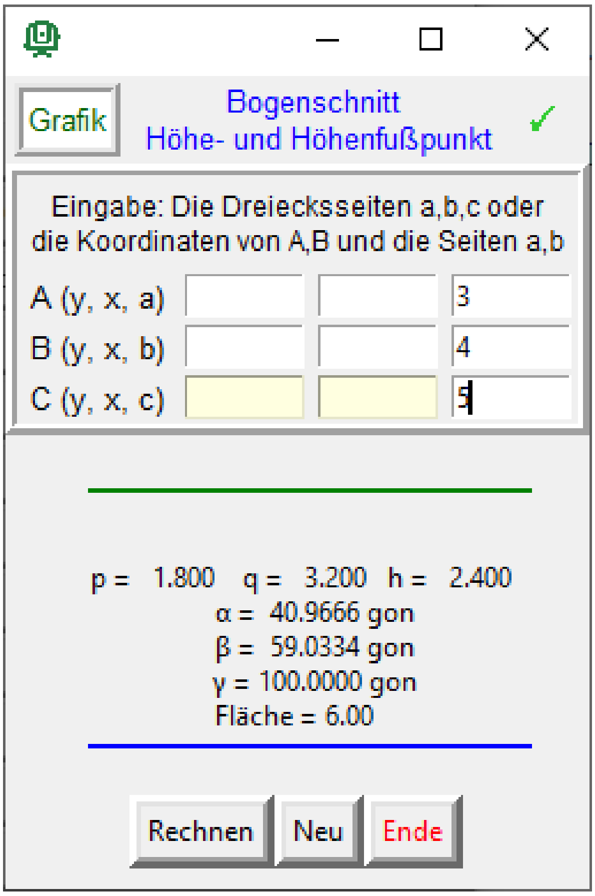


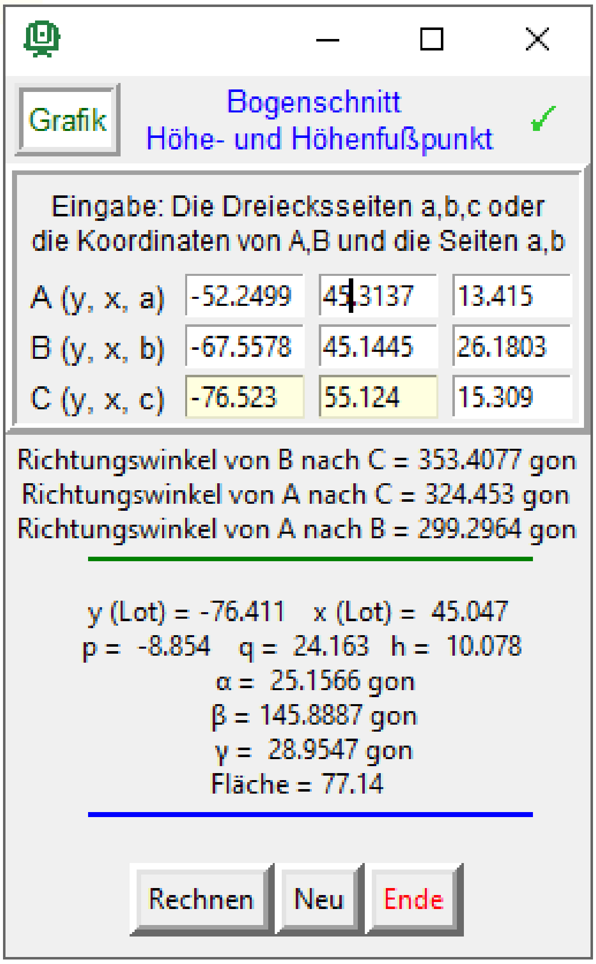

Gesucht:

$y_C, x_C$ Schnittpunkt C
$p, q, h$ Höhe und Höhenfußpunkt

Zusätzliche Ergebnisse:

$y_{Lot}, x_{Lot}$ Fußpunktkoordinaten
eventuell auch $c$
$\alpha, \beta, \gamma$ Dreieckswinkel [gon]
*Fläche*



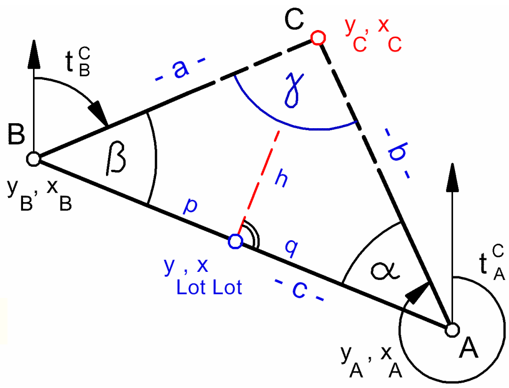

Eine mögliche Berechnung mit $y_A, x_A, y_B, x_B, a, b$ (Koordinaten und Seiten) ist vorrangig gegenüber der Berechnung alleinig aus den Dreiecksseiten $a, b, c$ .

# 13 Vorwärtsschnitt

### über Dreiecks- oder Richtungswinkel

Gegeben:

$y_\mathrm{A}, x_\mathrm{A}, y_\mathrm{B}, x_\mathrm{B}$ Koordinaten
$\alpha, \beta$ Dreieckswinkel [gon]
oder
$t_\mathrm{A}^\mathrm{C}, t_\mathrm{B}^\mathrm{C}$ Richtungswinkel [gon]

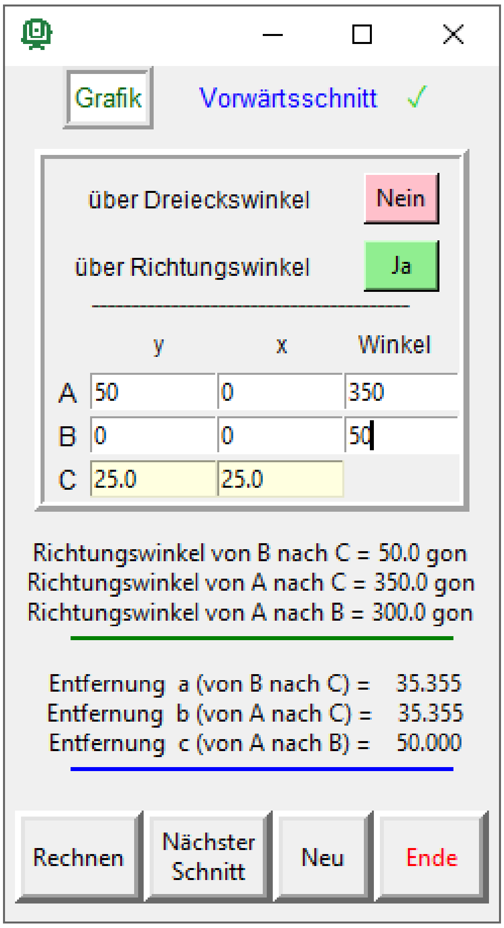


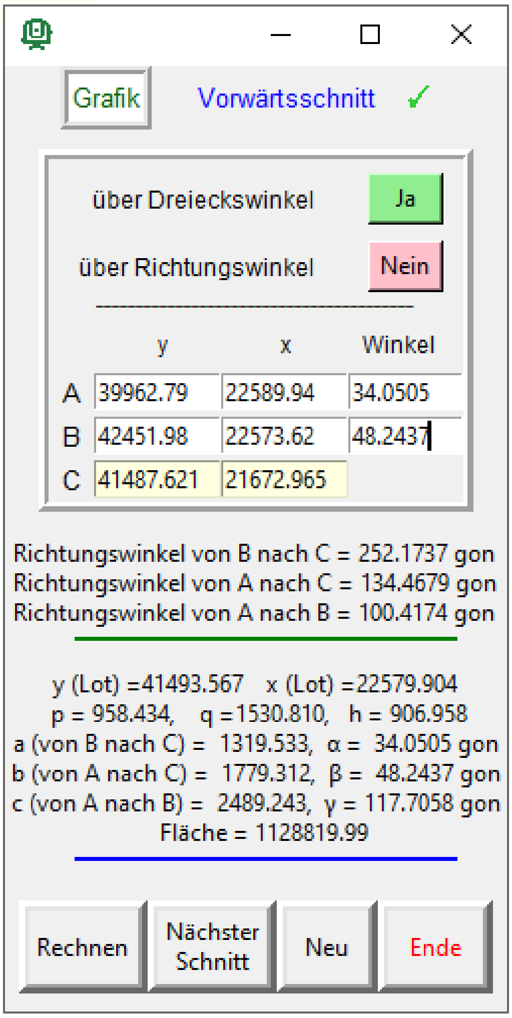

Gesucht:

$y_\mathrm{C}, x_\mathrm{C}$ Schnittpunkt C

Zusätzliche Ergebnisse:

$\alpha, \beta, \gamma$ Dreieckswinkel [gon]
$a, b, c$ (Basis) Dreiecksseiten [gon]
$p, q, h$ Höhe und Höhenfußpunkt
$y_\mathrm{Lot}, x_\mathrm{Lot}$ Fußpunktkoordinaten
*Fläche*



## 14 Pythagorasprobe

Gegeben:

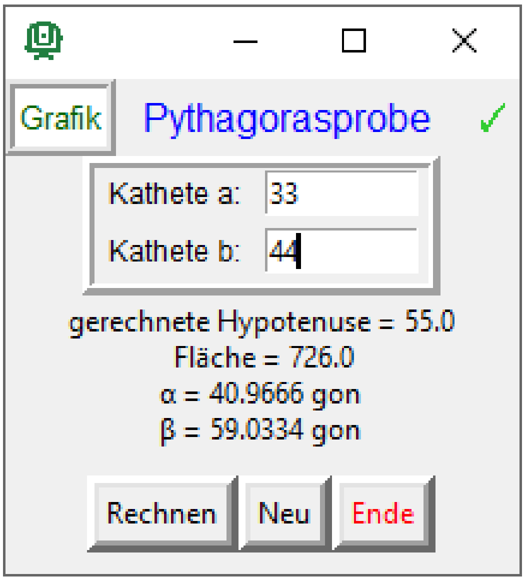
$a, b$ Katheten

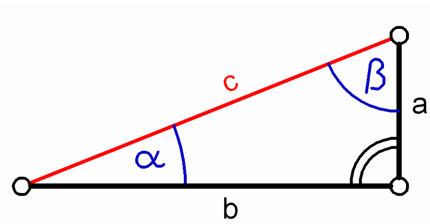
Gesucht:

$c$ Hypotenuse

Zusätzliche Ergebnisse:

*Fläche*
$\alpha, \beta$ Dreieckswinkel



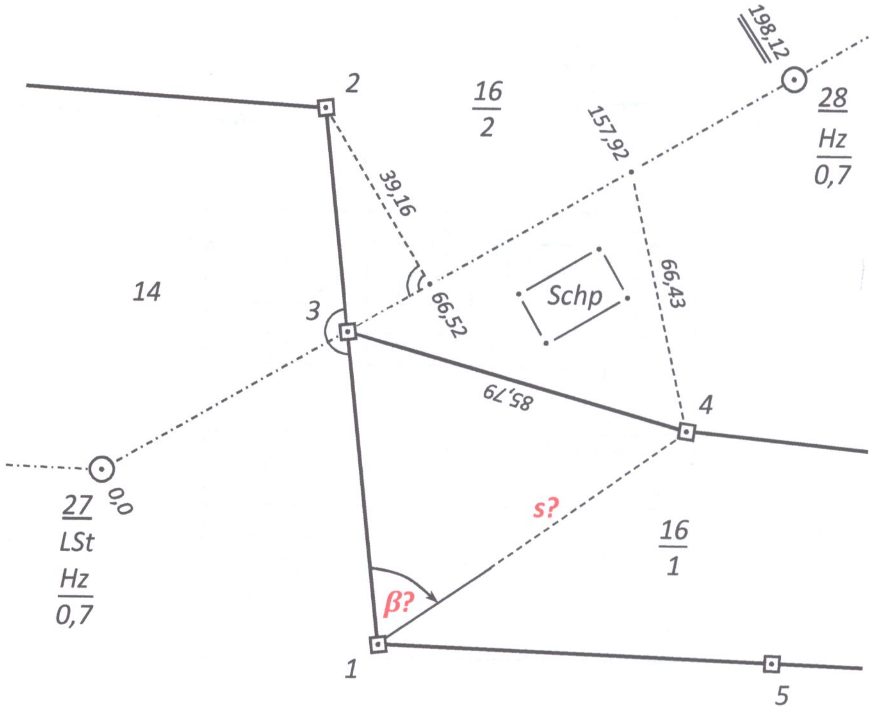

---

Das Programm wurde getestet mit

Samsung Galaxy Tab S4 – Android Vers. 10, python 3.9.7 Oct 6 2021
Samsung Galaxy A52s 5G (1080x2400 Pixel) – Android Vers. 12
Samsung Galaxy J5 (2015) (720x1480 Pixel) – Android Vers. 7.1.1
Windows 10 Pro 21H1

## Nachtrag: Anhang Beispiel

Das Beispiel entstammt aus dem Buch von **Sieland, Sammlung vermessungstechnischer Aufgaben mit ausführlichen Lösungen**, 3. Auflage 2022, Verlag Wichmann, Heidelberg: Beispiel Aufgabe 9.3 auf Seite 84. Lösung der Aufgabe daselbst auf Seite 199f.

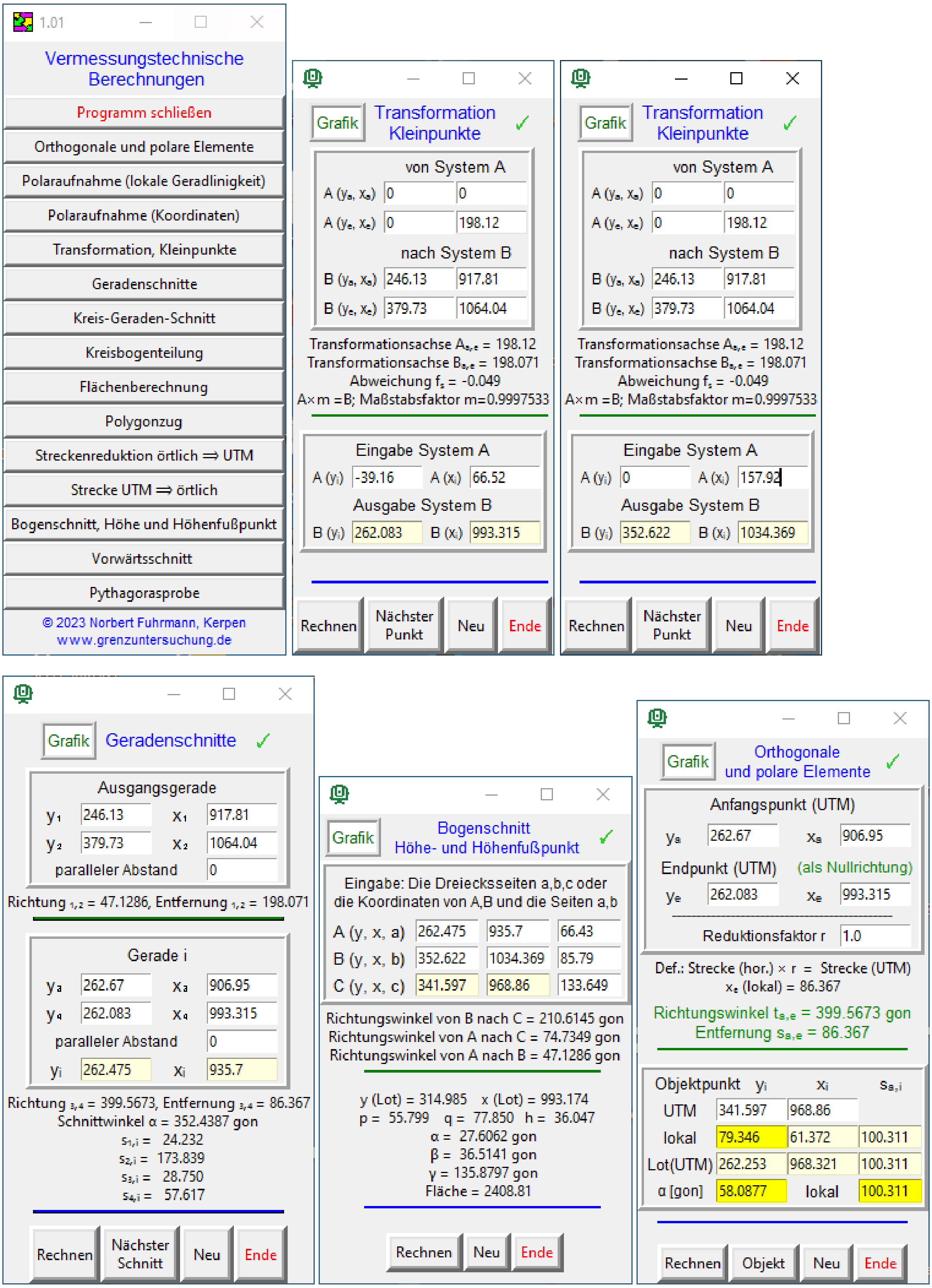



In der Örtlichkeit wurden nur die Grenzpunkte 1 und 2 vorgefunden. Vom Standpunkt 1 soll der Grenzpunkt 4 wiederhergestellt werden. Er soll polar abgesteckt werden; als Nullrichtung ist der Punkt 2 zu verwenden.

| Rechts | Hoch | Punkt-Nr |
|---|---|---|
| 246.13 | 917.81 | 27 |
| 379.73 | 1064.04 | 28 |
| 262.67 | 906.95 | 1 |

Die polaren Absteckungselemente sind zu berechnen.

Das Beispiel (s. folgende Seite) wurde in Windows mit zugleich geöffneten Programminstanzen (Tasks) berechnet. Gleiche oder unterschiedliche Instanzen der Programmteile können gleichzeitig geöffnet, darin gerechnet oder geschlossen werden. Daten können wahlweise mit Strg-C bzw. Strg-V in verschiedene Instanzen übertragen werden.




Ergebnis der Aufgabe 9.3: Standpunkt: 1

$\beta_2 = 0,0000 gon, s_2 = 86,367 m$

$\beta_4 = 58,0877 gon, s_4 = 100,311 m$