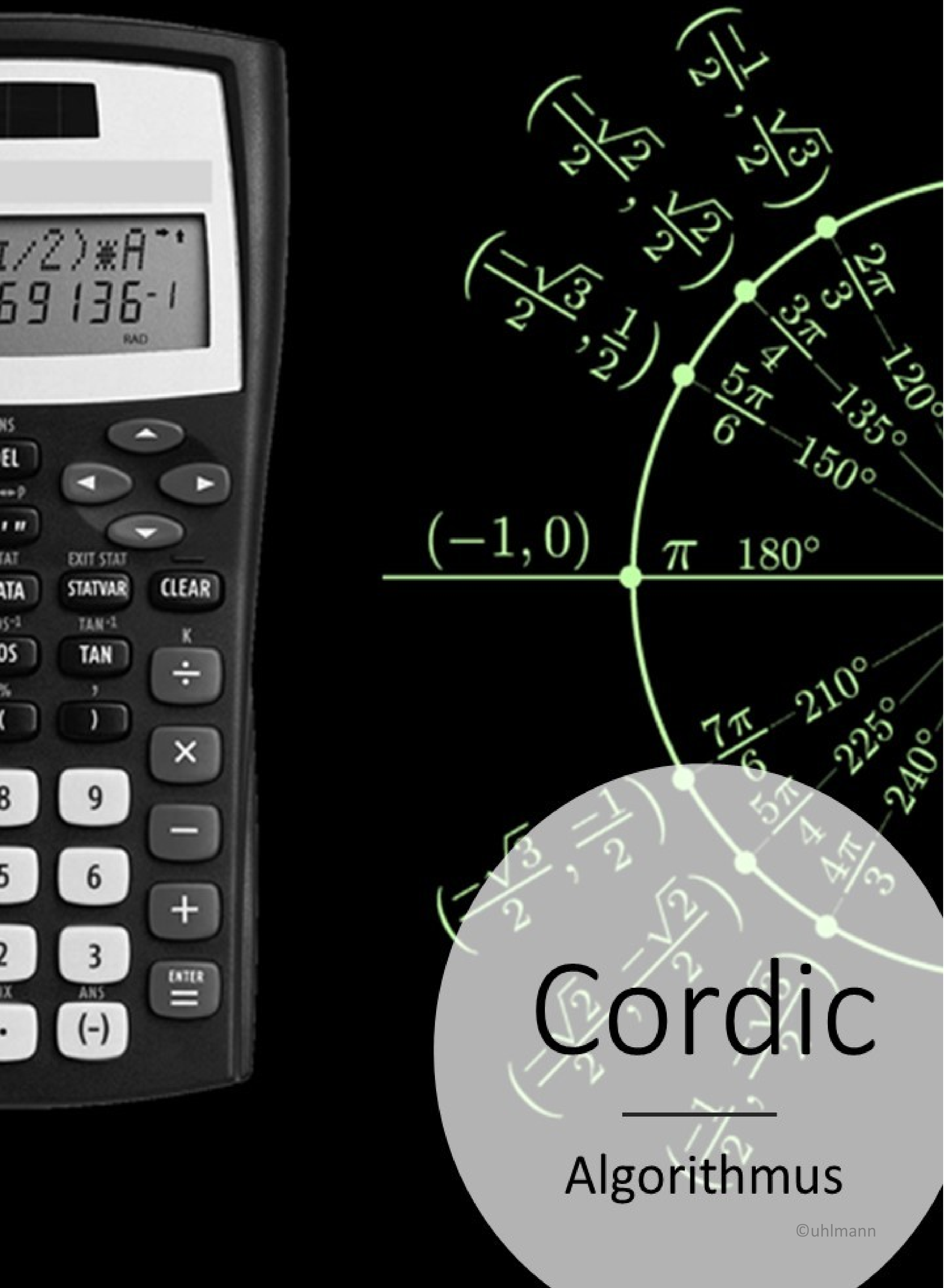

Cordic

Algorithmus

CORDIC - COordinate Rotation DIgital Computer

Für die schnelle Berechnung von Sinus- und Cosinuswerten konvergiert Taylor zu schlecht. Zur schnellen maschinellen Berechnung von Sinus und Cosinus für alle Winkel $0° < \varphi < 90°$ verwenden Taschenrechner und Computer den CORDIC-Algorithmus.

Vorübung: Iterationsverfahren zur Auffindung eines Winkels

Ein ganzzahliger Winkel $0 < \alpha < 90°$ soll in möglichst wenigen Schritten erraten werden. Als Rückmeldungen sind nur „größer" oder „kleiner" und natürlich „erraten" erlaubt.

Die Idee: Drehung des Punktes $A = (1,0)$ um den Winkel φ

Drehung von $A = (1,0)$ um den Winkel φ ergibt den Punkt $P = (\cos\varphi, \sin\varphi)$ am Einheitskreis!

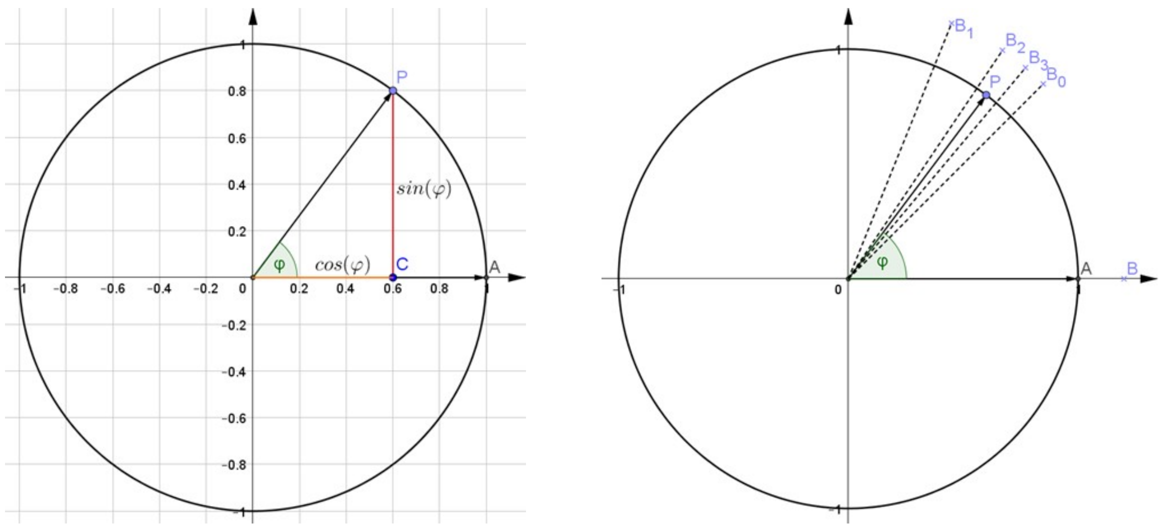

Abbildung 1

Die Drehung kann über eine Drehmatrix berechnet werden.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Da die Drehmatrix ihrerseits $\sin\varphi$ und $\cos\varphi$ benötigen, müssen wir anders vorgehen.

Die Idee: Wir drehen von $0°$ aus in aneinandergereihten, immer kleiner werdenden Schritten α_i, bis wir φ ausreichend genau erreichen:

$$P = \begin{pmatrix} x_n \approx \cos\varphi \\ y_n \approx \sin\varphi \end{pmatrix} = \ldots \cdot D_3 \cdot D_2 \cdot D_1 \cdot D_0 \cdot \begin{pmatrix} x_0 = 1 \\ y_0 = 0 \end{pmatrix}$$

$$\varphi = \sum_{i=0}^{\infty} \alpha_i \quad \begin{matrix} \alpha_i > 0 \ldots \mathit{Linksdrehung} \\ \alpha_i < 0 \ldots \mathit{Rechtsdrehung} \end{matrix}$$

z.B.: $\alpha_i = 45°, \pm 22.5°, \pm 11.25°, \pm 5.625°, \ldots \; \alpha_{i+1} = \pm \alpha_i / 2$ mit $\alpha_0 = 45°$.

Ideal $|\alpha_{i+1}| \approx |\alpha_i/2|$, wichtig aber $|\alpha_i/2| < |\alpha_{i+1}| \leq |\alpha_i|$ (Konvergenzkriterium)!

Beispiel: $\varphi = 53°$
Wir bleiben der Einfachheit ganzzahlig: $\alpha_0 = 45°, \alpha_1 = \pm 23°, \alpha_2 = \pm 12°, \alpha_3 = \pm 6°, \ldots$

$0° + 45° + 23° - 12° - 6° + 3° = \varphi = 53°$

Bei der Iteration werden 3 Variable protokolliert: x_i, y_i, z_i

x_i:
y_i: $\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} \cos\alpha_i & \mp\sin\alpha_i \\ \pm\sin\alpha_i & \cos\alpha_i \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$ mit $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

z_i: $z_{i+1} = z_i - \alpha_i$ mit $z_0 = \varphi$ (führt zu $z_i \rightarrow 0$)

Die Variable z_i bestimmt, ob der nächsten Schritt nach links oder rechts erfolgen muss.
Dabei werden für die Berechnung von z_i, ausgehend vom gesuchten Winkel $z_0 = \varphi$, die Schritte in umgekehrter Richtung durchgeführt, bis man annähernd bei Null angelangt ist.
Solange $z_i > 0$ bleibt, muss in der eigentlichen Iteration $\alpha_i > 0$ sein (Linksdrehung). Wird $z_i < 0$ muss auch $\alpha_i < 0$ gewählt werden (Rechtsdrehung).

Pseudo-Rotation und Korrekturfaktor

Der Algorithmus lässt sich in Abhängigkeit von nur einer Winkelfunktion darstellen.

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \cos\alpha_i \cdot \begin{pmatrix} 1 & \mp\tan\alpha_i \\ \pm\tan\alpha_i & 1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \underbrace{\frac{1}{\sqrt{1+(\tan\alpha_i)^2}}}_{k_i} \cdot \underbrace{\begin{pmatrix} 1 & \mp\tan\alpha_i \\ \pm\tan\alpha_i & 1 \end{pmatrix}}_{R(\alpha_i)} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

Die Rotation durch die jeweilige Drehmatrix D_i spaltet sich dabei in zwei Teile $D_i = k_i \cdot R(\alpha_i)$. Die eigentliche Drehung erfolgt durch $R(\alpha_i)$. Der Betrag des Ergebnisvektors ist dann aber größer als 1, der Vektor ragt also über den Einheitskreis hinaus (Pseudo-Rotation). Der Faktor $k_i < 1$ normiert den Vektor wieder.

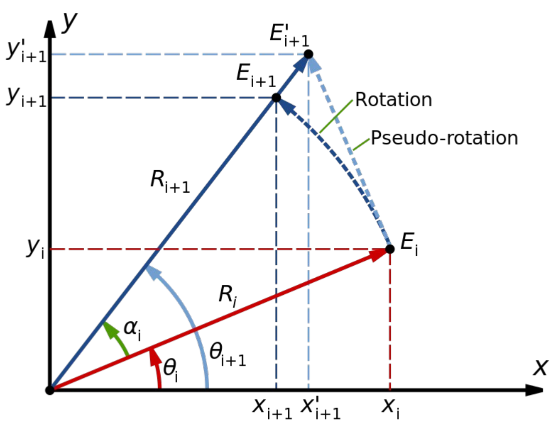

Abbildung 2

In der Iteration führen die wiederholten Drehungen D_i dann zu

$$P=\begin{pmatrix} x_n \\ y_n \end{pmatrix}=\begin{pmatrix} x_n \approx \cos\varphi \\ y_n \approx \sin\varphi \end{pmatrix}=\ldots\cdot D_3\cdot D_2\cdot D_1\cdot D_0\cdot\begin{pmatrix} x_0=1 \\ y_0=0 \end{pmatrix}=\text{¿}$$

$$\text{¿}\left[\prod_{i=0}^{n} k_i\cdot R(\alpha_i)\right]\cdot\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}=\prod_{i=0}^{n} k_i\cdot\left[\underbrace{\prod_{i=0}^{n} R(\alpha_i)\cdot\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}}_{\text{PseudoRotation}}\right]$$

$$\text{wobei}\prod_{i=0}^{n} k_i \rightarrow K\,\text{konvergiert}\,! \rightarrow \prod_{i=0}^{\infty} k_i=K$$

Vereinfachung der Pseudo-Rotationsmatrix

Um den Algorithmus an die einfachen Bauteile eines Mikrokontrollers (z.B. Taschenrechner) anzupassen, werden die benötigten Tangenswerte durch einigermaßen passende binäre Werte ersetzt. Dabei werden nicht die Drehwinkel, sondern die Tangenswerte halbiert.

α_i–Tabelle

α_i	$\tan\alpha_i$
$\alpha_0=45°$	$1{,}00000\ldots$
$\alpha_1=22{,}5°$	$0{,}41421\ldots$
$\alpha_2=11{,}25°$	$0{,}19891\ldots$
$\alpha_3=5{,}625°$	$0{,}09849\ldots$
$\ldots$	$\ldots$

Alternative α_i – Tabelle

α_i	$\tan\alpha_i$
$\alpha_0=45°$	$1{,}0=2^{-0}$
$\alpha_1=26{,}565\ldots°$	$0{,}5=2^{-1}$
$\alpha_2=14{,}036\ldots°$	$0{,}25=2^{-2}$
$\alpha_3=7{,}125\ldots°$	$0{,}125=2^{-3}$
$\ldots$	$\ldots$

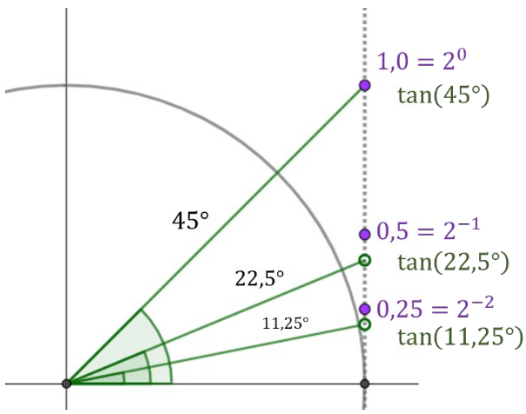

Abbildung 3: Tangenshalbierung versus Winkelhalbierung

Die Pseudo-Rotationsmatrix $R(\alpha_i)$ wird durch die Tangensmatrix R_i ersetzt.

$$R(\alpha_i)\rightarrow R_i=\begin{pmatrix} 1 & \mp 2^{-i} \\ \pm 2^{-i} & 1 \end{pmatrix},\ k_i=\frac{1}{\sqrt{1+2^{-2i}}}$$

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix}=k_n R_n\cdot\ldots\cdot k_3 R_3\cdot k_2 R_2\cdot k_1 R_1\cdot k_0 R_0\cdot\begin{pmatrix} 1 \\ 0 \end{pmatrix}=\prod_{i=0}^{n} k_i\cdot\prod_{i=0}^{n} R_i\cdot\begin{pmatrix} x_i \\ y_i \end{pmatrix}\cong\begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \end{pmatrix}$$

Iteration in 3 Variablen

$x_n \rightarrow \cos\varphi\,,\, y_n \rightarrow \sin\varphi\,,\, z_n \ldots$Winkelkontrollvariable

$x_0=1$	$x_{i+1}=x_i-\sigma_i\cdot 2^{-i}\cdot y_i$	$\sigma_i=\operatorname{sgn}(z_i)=\begin{cases}1 & z_i>0\\ 0 & z_i=0\\ -1 & z_i<0\end{cases}$
$y_0=0$	$y_{i+1}=\sigma_i\cdot 2^{-i}\cdot x_i+y_i$	
$z_0=\varphi$	$z_{i+1}=z_i-\sigma_i\cdot \underbrace{\tan^{-1}(2^{-i})}_{\alpha_i}$	α_i … lookup table*

Bis $|z_i|<$¿ dem vorgegebenen Toleranzwert ≈ 0 oder $x_{i+1}-x_i\approx 0$ bzw. $y_{i+1}-y_i\approx 0$

Die konkreten Werte für $\tan^{-1}(2^{-i})$ werden dabei tabellarisch im Rechenspeicher hinterlegt.

Das Ergebnis der Pseudo-Rotationen R_i muss am Ende noch mit dem Skalierungsfaktor

$K=\prod_{i=0}^{\infty} k_i=\prod_{i=0}^{\infty}\frac{1}{\sqrt{1+2^{-2i}}}=0{,}60725\ldots$multipliziert werden!

Zusammenfassung:

$$\begin{pmatrix}1\\0\end{pmatrix}\rightarrow\begin{pmatrix}x'_n\\y'_n\end{pmatrix}=\prod_{i=0}^{n}R_i\cdot\begin{pmatrix}x_i\\y_i\end{pmatrix}\rightarrow\begin{pmatrix}x_n\\y_n\end{pmatrix}=K\cdot\begin{pmatrix}x'_n\\y'_n\end{pmatrix}\approx\begin{pmatrix}\cos\varphi\\\sin\varphi\end{pmatrix}$$

Erweiterung:

In etwas abgeänderter Form erlaubt die Iteration auch, den $\sin^{-1}(\upsilon)$ zu berechnen. Dabei wird der Anfangsvektor soweit gedreht, bis die y-Koordinate des gedrehten Vektors mit dem Wert von υ übereinstimmt. Die dabei aufakkumulierten Teilwinkel ergeben dann den gesuchten Winkel $\sin^{-1}(\upsilon)$. Daraus ergeben sich in weiterer Folge auch $\cos^{-1}\square$ und $\tan^{-1}\square$.

In einer verallgemeinerten Form eignet sich der CORDIC-Algorithmus auch zur Berechnung der hyperbolischen Funktionen sowie der Exponential- und Logarithmusfunktion.

Beispiel $\varphi = 29°$

x_0: y_0:	$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
z_0:	$z_0 = 29°;$

x_1: y_1:	$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 & -2^{-0} \\ +2^{-0} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} 1 & -2^{-0} \\ +2^{-0} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
z_1:	Drehwinkel $\alpha_0 = \tan^{-1}(2^{-0}) = ¿45° \rightarrow z_1 = z_0 - \alpha_0 = 29° - 45° = -16°¿$

x_2: y_2:	$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & +2^{-1} \\ -2^{-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 & +2^{-1} \\ -2^{-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1{,}5 \\ 0{,}5 \end{pmatrix}$
z_2:	Drehwinkel $\alpha_1 = ¿$

x_3: y_3:	$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & -2^{-2} \\ +2^{-2} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & -2^{-2} \\ +2^{-2} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1{,}5 \\ 0{,}5 \end{pmatrix} = \begin{pmatrix} 1{,}375 \\ 0{,}875 \end{pmatrix}$
z_3:	Drehwinkel $\alpha_2 = \tan^{-1}(2^{-2}) = ¿14{,}036° \rightarrow z_3 = z_2 - \alpha_2 = 10{,}565° - 14{,}036° = -3{,}471°¿$

x_4: y_4:	$\begin{pmatrix} x_4 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & +2^{-3} \\ -2^{-3} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & +2^{-3} \\ -2^{-3} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1{,}375 \\ 0{,}875 \end{pmatrix} = \begin{pmatrix} 1{,}484 \\ 0{,}703 \end{pmatrix}$
z_4:	Drehwinkel $\alpha_3 = ¿$

x_5: y_5:	$\begin{pmatrix} x_5 \\ y_5 \end{pmatrix} = \begin{pmatrix} 1 & -2^{-4} \\ +2^{-4} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_4 \\ y_4 \end{pmatrix} = \begin{pmatrix} 1 & -2^{-4} \\ +2^{-4} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1{,}484 \\ 0{,}703 \end{pmatrix} = \begin{pmatrix} 1{,}440 \\ 0{,}796 \end{pmatrix}$
z_5:	Drehwinkel $\alpha_4 = \tan^{-1}(2^{-4}) = ¿7{,}125° \rightarrow z_5 = z_4 - \alpha_4 = 3{,}654° - 7{,}125° = 0{,}078°¿$

ABBRUCH! $z_i \cong 0$

LÖSUNG: $\begin{pmatrix} \cos 29° \\ \sin 29° \end{pmatrix} \cong 0{,}60725 \cdot \begin{pmatrix} 1{,}440 \\ 0{,}796 \end{pmatrix} = \begin{pmatrix} 0{,}8747 \\ 0{,}4833 \end{pmatrix}$

Die tatsächlichen Werte sind: $\cos 29° = 0{,}8746$ und $\sin 29° = 0{,}4848$