

Troepfel

Georg Pfundt

Copyright © Â©byGeorgPfundt03-JAN-1996 16:05:21

COLLABORATORS

	<i>TITLE :</i> Troepfel		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Georg Pfundt	June 8, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Troepfel	1
1.1	Tröpfel-Algorithmus für Pi	1
1.2	Die Einführung	1
1.3	Der Algorithmus	2
1.4	Rechenzeit	2
1.5	Literatur	2
1.6	Der Preis	3

Chapter 1

Troepfel

1.1 Tröpfel-Algorithmus für Pi

Der Tröpfelalgorithmus für Pi

© by Georg Pfundt

03-JAN-1996

E-MAIL:

gp@ict.fhg.de (bis 30-APR-1996)

pfundt@fphvax.rz.uni-karlsruhe.de (bis 31-DEZ-1996)

Worum geht's? [Die Einführung](#)

Wie funktioniert's? [Der Algorithmus](#)

Wie schnell ist's? [Die Rechenzeit](#)

Wo kann man's nachlesen? [Die Literatur](#)

Was kostet's? [Der Preis](#)

1.2 Die Einführung

Einführung

Das Programm TROEPFEL berechnet mit dem sogenannten [Tröpfel Algorithmus](#) beliebig viele Stellen von Pi bzw. e . Der Programmaufruf ist denkbar einfach. Vom CLI aus gibt man ein:

```
troepfel pi 100
```

und geschwind erscheinen die ersten hundert Stellen von PI auf dem Bildschirm.

```
3.1415926535897932384626433832795028841971693993751
```

```
05820974944592307816406286208998628034825342117067
```

Gibt man nach der Anzahl der Signifikanten Stellen noch einen Dateinamen an, wird das Ergebnis direkt in eine Datei geschrieben.

Es stehen zwei Versionen des Programms zur Verfügung:

Troepfel000 Für antike 68000'er Amigas.

Troepfel020 Für 68020+ Amigas.

Die 68020 Version ist etwa 35% Schneller als die 68000'er. Ein Gleitkommaprozessor (68881) wird nicht benötigt, da das Programm nur ganzzahlige Berechnungen durchführt.

Da das Programm die Utility.library von Commodore benutzt, benötigt man mindestens die Betriebssystemversion 2.0 um das Programm zu starten. Der Quellcode sollte sich aber auf allen ANSI-Kompatiblen C-Compilern und Betriebssystemen übersetzen lassen.

Zum guten Schluß noch Pi mit 2000 und e mit 4000 Stellen Genauigkeit.

1.3 Der Algorithmus

Der Tröpfel Algorithmus für Pi

Der Algorithmus gelangte zu seinem Namen, da das Ergebnis während der Berechnung Stück für Stück "heraustropft" und nicht erst am Ende der Berechnung zur Verfügung stehen, wie bei anderen Verfahren.

Das Prinzip ist einfach: Man stellt die gesuchte Zahl nicht im Zehnersystem sondern in einer geeigneteren Basis dar und plötzlich sieht es viel einfacher aus. Als Basis wählt man nicht wie üblich eine Potenzbasis wie a^0 , a^1 , a^2 , ..., sondern z.B für Pi $1/3$, $2/5$, $3/7$, $4/9$, Damit läßt sich Pi dann als $2,2222222...$ schreiben. Jetzt muß man nur noch Pi ins Zehnersystem umrechnen und schon hat man das Ergebnis.

Diese Umrechnung erreicht man indem man die ganze Zahl jeweils mit 10 multipliziert und den links vor das Komma rutschenden ganzzahligen Teil abtrennt und ausgibt.

Das wars.

1.4 Rechenzeit

Die Rechenzeit

Die Rechenzeit wächst quadratisch mit der Anzahl der signifikanten Stellen: Also doppelt so viele Stellen -> vierfache Rechenzeit!

Auf meinem Amiga 4000/040 mit 25Mhz sind für Pi also maximal einige 10000 Stellen in vernünftiger Zeit zu berechnen. Benötigt man mehr, sollte man den Algorithmus von Ramanujan oder besser von Borwein und Borwein implementieren. Allerdings benötigen diese Verfahren auch Arithmetikroutinen für entsprechend lange Zahlen. Hat man diese zur Hand, so ist der Rest ein Kinderspiel.

Die Berechnung von e läuft ca. drei mal schneller als die von pi und benötigt auch weniger Speicher (aber das ist ja eh nicht viel). Für e braucht man $2*n$ Bytes für n signifikante Stellen, für Pi sind es $\text{Floor}(10*n/3)*4$ Bytes also ca. $13.3*n$ Bytes.

Hier noch die Rechenzeiten für die beiden Beispieldateien von Pi mit 2000 und e mit 4000 Stellen Genauigkeit:

e: 80.8s -> 49.5 Stellen/s

Pi: 78.2s -> 25.6 Stellen/s

1.5 Literatur

Literatur zu dem Thema

Das Programm ist nach einem Artikel im Spektrum der Wissenschaft, Dezember 1995, S. 10-14 von Ian Stewart geschrieben. Als weiterführende Literaturangaben sind dort u.A. enthalten:

"A Spigot Algorithm for the Digits of Pi", Rabinowitz, Stanley; Wagon, Stan; American Mathematical Monthly, v102 n3 a1995 pp195-203

"The Calculation of e to many Significant Digits", Sale, A. H. J.; Computing Journal, v11 a1968 pp229-230

1.6 Der Preis

Alles hat seinen Preis...

...nur dieses Programm kostet nichts! Das Copyright für diese Version des Programmes liegt bei mir. Der Quellcode und das Programm dürfen außer zu militärischen oder zu menschenrechtsverletzenden Zwecken nach belieben kopiert und verwendet werden.