Pi Day im Computermuseum

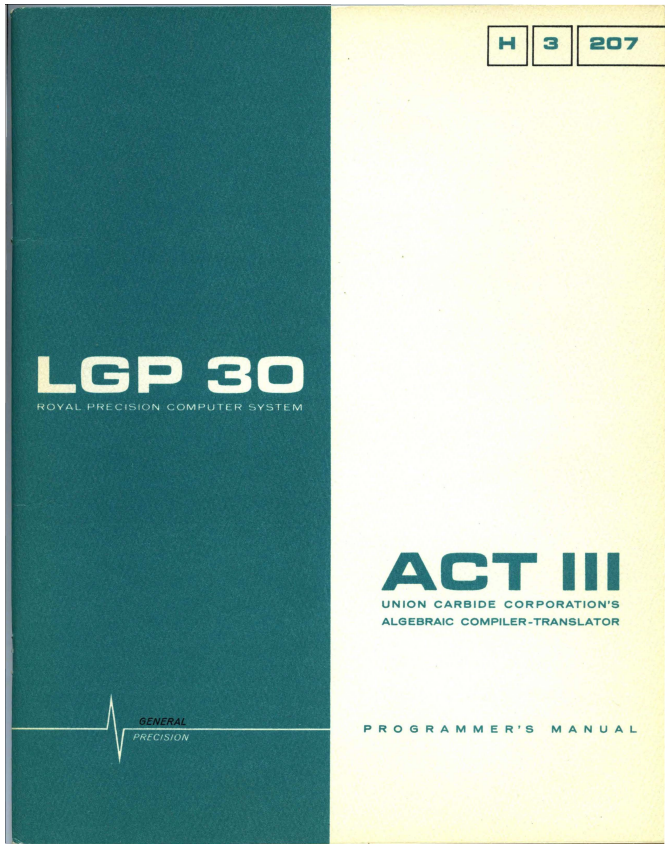
Teil 3

Jan Köllner

3-14-2025

ACT-III und ACT-V

- Programmiersprache speziell für den LGP-30, Leistungsumfang ähnlich zu Fortran II.
- Integerbereich:
  $\pm 536\,870\,911 = \pm(2^{29} - 1)$
  (vgl. Wortbreite von 32 Bit)
- Floating-Point: $y = a \cdot 10^m$ mit
  $0.100\,000\,02 \leq |a| \leq 0.999\,999\,94$
  und $-32 \leq m \leq 31$
- Variablen, Arrays, Verzweigungen, Sprünge, Schleifen, Ein- und Ausgaberoutinen



Reihendarstellung für π

- Madhava (1350-1425), James Gregory (1671) und Gottfried Wilhelm Leibniz (1673): Für $|x| \leq 1$ ist

$$\arctan(x) = \sum_{j=0}^{\infty} \frac{(-1)^j}{2j+1} x^{2j+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots$$

  dann $\arctan(1) = \pi/4$.
- Verbesserung von John Machin (1706): Mit

$$\arctan\left(\frac{x_1}{y_1}\right) + \arctan\left(\frac{x_2}{y_2}\right) = \arctan\left(\frac{x_1 y_2 + x_2 y_1}{y_1 y_2 - x_1 x_2}\right)$$

  falls $-\pi/2 < \arctan(x_1/y_1) + \arctan(x_2/y_2) < \pi/2$ folgt daraus

$$\frac{\pi}{4} = 4\arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right)$$

Reihendarstellung für π

- Madhava (1350-1425), James Gregory (1671) und Gottfried Wilhelm Leibniz (1673): Für $|x| \leq 1$ ist

$$\arctan(x) = \sum_{j=0}^{\infty} \frac{(-1)^j}{2j+1} x^{2j+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots$$

  dann $\arctan(1) = \pi/4$.

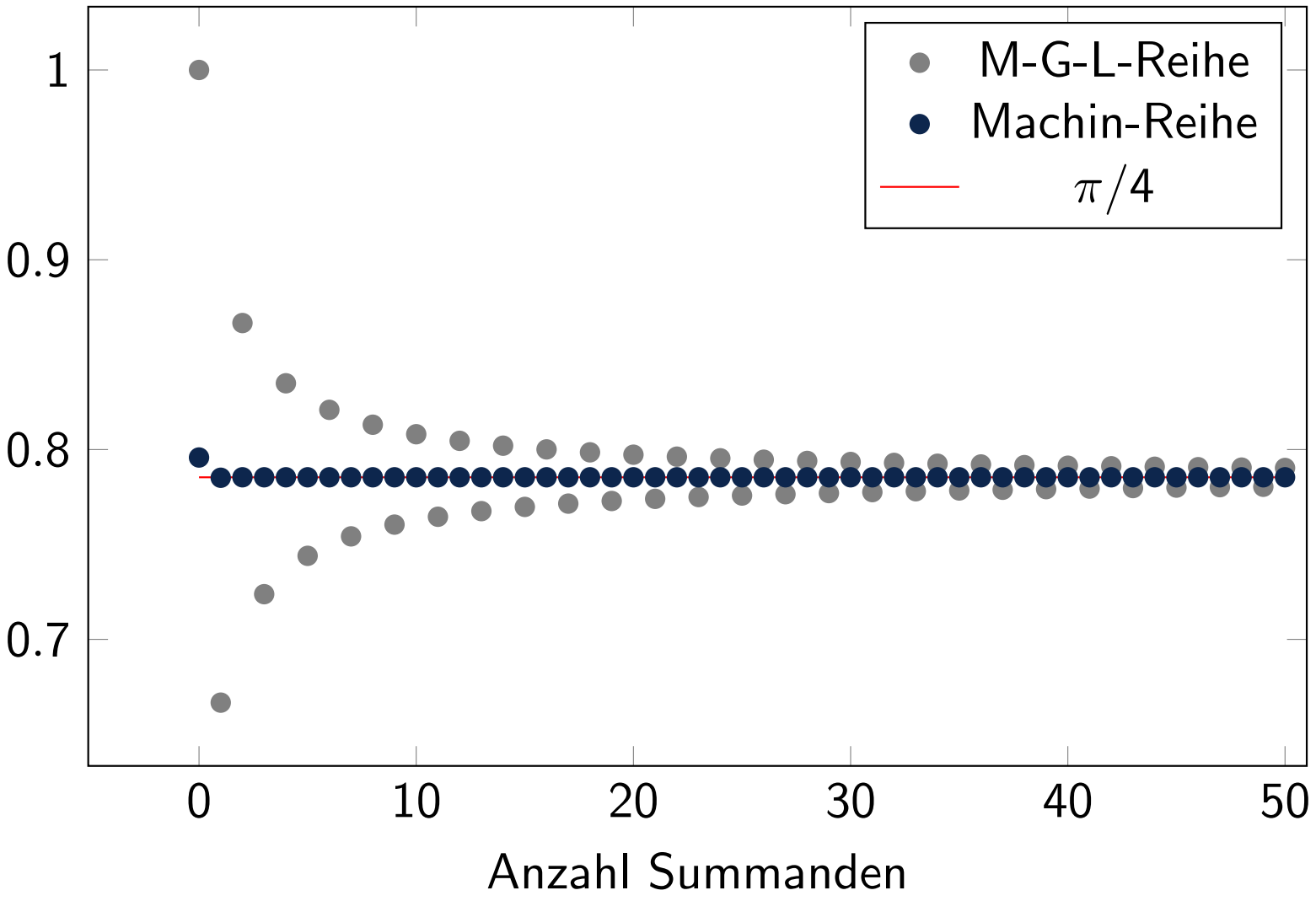
- Verbesserung von John Machin (1706): Mit

$$\arctan\left(\frac{x_1}{y_1}\right) + \arctan\left(\frac{x_2}{y_2}\right) = \arctan\left(\frac{x_1 y_2 + x_2 y_1}{y_1 y_2 - x_1 x_2}\right)$$

  falls $-\pi/2 < \arctan(x_1/y_1) + \arctan(x_2/y_2) < \pi/2$ folgt daraus

$$\begin{aligned} \frac{\pi}{4} &= 4\arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right) \\ &= 4 \cdot \sum_{j=0}^{\infty} \frac{(-1)^j}{2j+1} \left(\frac{1}{5}\right)^{2j+1} - \sum_{j=0}^{\infty} \frac{(-1)^j}{2j+1} \left(\frac{1}{239}\right)^{2j+1} \end{aligned}$$

Konvergenzvergleich



Implementierung

Berechne

$$\frac{\pi}{4} \approx 4 \cdot \sum_{j=0}^{45} \frac{(-1)^j}{2j+1} \left(\frac{1}{5}\right)^{2j+1} - \sum_{j=0}^{14} \frac{(-1)^j}{2j+1} \left(\frac{1}{239}\right)^{2j+1} \in]0, 1[$$

zur Basis 10^6, d.h.

$$\frac{\pi}{4} \approx \sum_{j=1}^{20} a_j \cdot 10^{-6j}$$

mit Ziffern $a_j \in [0, 999\,999]$. Diese speichern wir in einem Array:

a_1	a_2	a_3			$\cdots$			a_{19}	a_{20}

Wir verwenden mehrere Arrays: für die Potenzen, die Summanden und das Zwischenergebnis.
Für die Arrays implementieren wir Addition, Subtraktion, Multiplikation und Division mit entsprechenden Überträgen.

Der Code – Basics

```
    +1000'000';'base''
    dim'pow'2l''
    dim'pi'2l''
    dim'pis'2l''

    0';'j''
s0l'0';'pow'j''
    0';'pi'j''
    0';'pis'j''
    for'j'step'l'until'20'rpeat's0l''
```

⟵ der Variablen base wird der Wert 1 000 000 zugewiesen

Der Code – Basics

```
    +1000'000';'base''
```

← der Variablen `base` wird der Wert 1 000 000 zugewiesen

```
    dim'pow'21''
    dim'pi'21''
    dim'pis'21''
```

← drei Arrays (`pow`, `pi`, `pis`) der Länge 21 werden deklariert

```
    0';'j''
s01'0';'pow'j''
    0';'pi'j''
    0';'pis'j''
    for'j'step'1'until'20'rpeat's01''
```

Der Code – Basics

```
    +1000'000';'base''
```

← der Variablen `base` wird der Wert 1 000 000 zugewiesen

```
    dim'pow'21''
    dim'pi'21''
    dim'pis'21''
```

← drei Arrays (`pow`, `pi`, `pis`) der Länge 21 werden deklariert

```
    0';'j''
s01'0';'pow'j''
    0';'pi'j''
    0';'pis'j''
    for'j'step'1'until'20'rpeat's01''
```

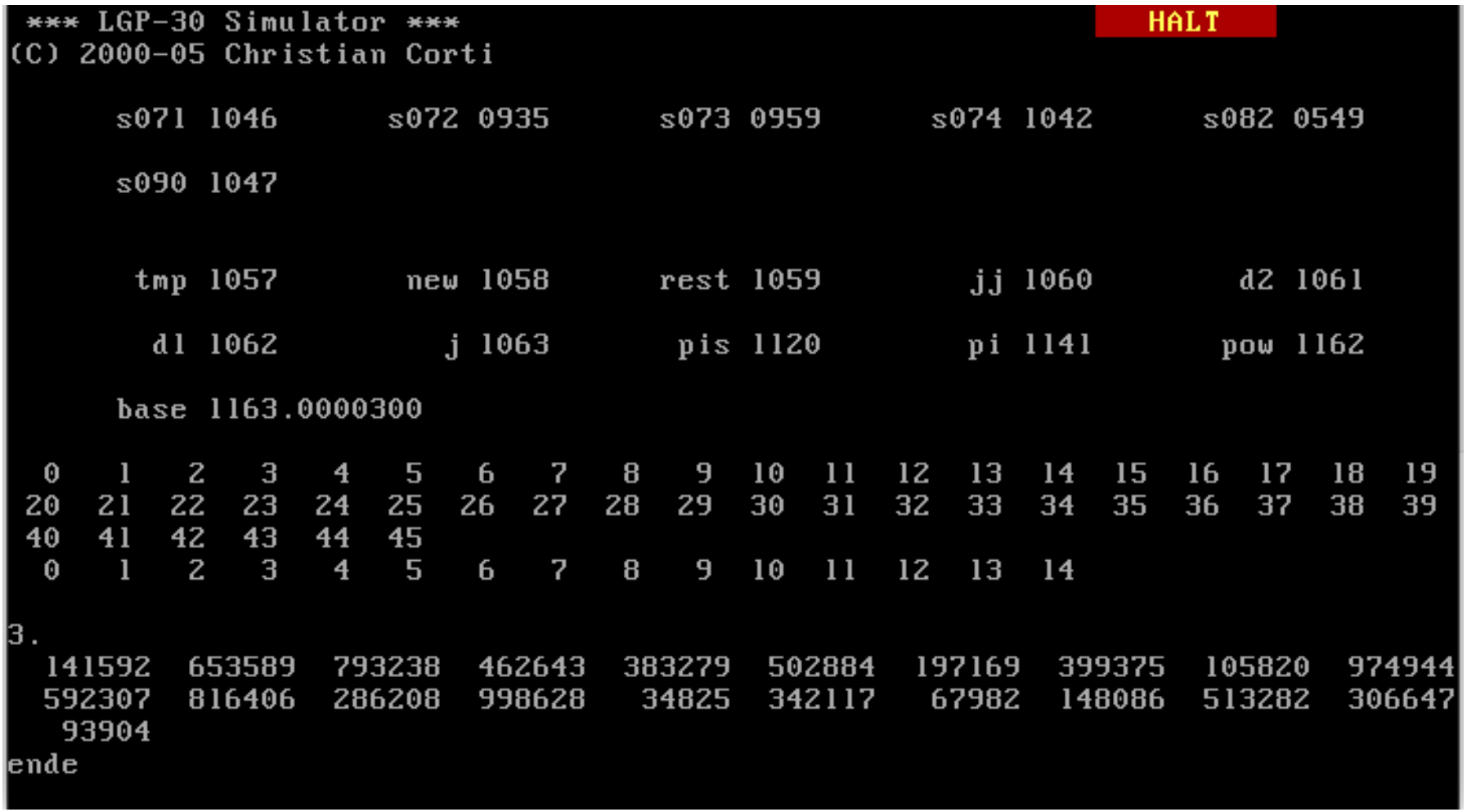
← Nullinitialisierung der drei Arrays mit For-Schleife

Der Code – Addition mit Übertrag

"Subroutine"für die Addition von `pis` zu `pi`:

```
sl0'20';'j''                          ⟵ beginne mit der letzten Ziffer
sl2'pi'j'i+'pis'j';'pi'j''            ⟵ komponentenweise Addition
    pi'j'i-'base';'tmp''
    if'tmp'neg'sl3''                  ⟵ gibt es einen Übertrag?
    tmp';'pi'j''
    j'i-'l';'j''                      ⟵ verarbeite Übertrag
    pi'j'i+'l';'pi'j''
    use'sl2''
sl3'j'i-'l';'j''
    if'j'zero'sl2'pos'sl2''           ⟵ bei erster Ziffer angekommen?
sll'go to's0''                        ⟵ Ausstieg aus Subroutine
```

Aufruf der Subroutine mit

```
ret'sll'use'sl0''
```

It works on my machine