

Application note 009, rev 1

Further guide to calculate programmed sums, demonstrated with the Euler-Mascheroni and the C47 Σ_n function. This is based a contribution made by C47 enthusiast Jozef Ongena, and written up by Jaco and illustrates the remarkable fast converging series for the Euler-Mascheroni constant:

This is a modification of the original Euler-Mascheroni formula by Brent:

$$\gamma = \lim_{n \rightarrow \infty} \frac{\sum_{k=0}^{\infty} (n^k/k!)^p (H_k - \ln(n))}{\sum_{k=0}^{\infty} (n^k/k!)^p}$$

Euler-Mascheroni ([Brent et al](#))(Eqn. 17)

$$H_n = \sum_{k=1}^n \frac{1}{k}$$

Harmonic [number](#),

The exponent p and the fixed value for n can be chosen in function of the precision required, as explained in Brent's referenced paper. For n in the formula, we used 30, which should allow a precision of more than 30 digits, but the accuracy is tested for below in the C47 program.

Jozef: "The advantage of the series by Brent is that the original definition of the Euler-Mascheroni constant ($\lim_{n \rightarrow \infty} (H_n - \ln(n))$) is very slowly converging. This (Brent) series is very fast. Only 100 terms have to be taken to get ~25 precise numbers! And it takes a minute or so connected to the power with the C47!"

Breaking down Brent's formula into numerator & denominator, as well as the Harmonic Number generator and the main program which sequences the finite sums:

```
C47 Program file export: Export
format version 2, C47 program
version 1.
0000: { Prgm #47: 65 bytes / 15 steps }
0001: LBL 'EulerM'
0002: 1
0003: 100
0004: 1
0005: SUMn 'denom'
0006: STO 01
0007: 1
0008: 100
0009: 1
0010: SUMn 'numer'
0011: STO 02
0012: RCL 01
0013: /
0014: 1/X
0015: .END.
```

Main program, numerator

```
C47 Program file export: Export format
version 2, C47 program version 1.
0000: { Prgm #45: 28 bytes / 11 steps }
0001: LBL 'numer'
0002: 30
0003: RCL T
0004: Y^X
0005: RCL T
0006: X!
0007: /
0008: 2
0009: Y^X
0010: RTN
0011: END
```

```
C47 Program file export: Export format
version 2, C47 program version 1.
0000: { Prgm #44: 12 bytes / 4 steps }
0001: LBL 'HarmN'
0002: 1/X
0003: RTN
0004: END
```

Harmonic number calc

```
C47 Program file export: Export format
version 2, C47 program version 1.
0000: { Prgm #46: 59 bytes / 21 steps }
0001: LBL 'denom'
0002: 30
0003: RCL T
0004: Y^X
0005: RCL T
0006: X!
0007: /
0008: 2
0009: Y^X
0010: STO 11
0011: 1
0012: RCL T
0013: 1
0014: SUMn 'HarmN'
0015: 30
0016: LN
0017: -
0018: RCL 11
0019: *
0020: RTN
0021: END
```

Denominator

C47 results:

PGMSLV	SLVC	f''(x)	iΠ _n	iΣ _n	PGMINT
SOLVE	SLVQ	f'(x)	Π _n	Σ _n	∫fdx

Y: Result of the above programmed series

X: Built-in constant γ_{EM}

Subtracting the built-in Euler-Mascheroni constant from the series result, the resulting accuracy for $p = 2$ is shown to be $6.749849059 \times 10^{-25}$.

Raising the arbitrarily chosen p from 2 to 4 (line 8 of [denom] and [numer]) increases the accuracy to 1×10^{-34} , see the table below.

p	n	Number of terms	Proximity to the Euler Constant
2	30	100	$6.749849059 \times 10^{-25}$
4	30	100	-1×10^{-34}

Source for the example: MATHEMATICS OF COMPUTATION, VOLUME 34, NUMBER 149, JANUARY 1980, PAGES 305-312, *Some New Algorithms for High-Precision Computation of Euler's Constant*, [link](#)

[Author Jaco Mostert & Jozef Ongena](#)

Copyright (C) 2024 The C47 Development Team. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be downloaded from gnu.org.