Application note 009

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:

$$\gamma = \lim_{n\to\infty} \frac{\sum_{k=0}^{\infty} (n^k/k!)^p (H_k - \ln(n))}{\sum_{k=0}^{\infty} (n^k/k!)^p}. \qquad H_n = \sum_{k=1}^{n} \frac{1}{k}$$

Euler-Mascheroni (Brent et al)(Eqn. 17) *Harmonic number,*

Jozef: “The advantage of the series by Brent is that the original definition of the Euler-Mascheroni constant `(lim n->inf (Hn - 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!

For n in the formula, we took 30, this should allow a precision of more than 30 digits.”

Breaking down Brent’s formula in its 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

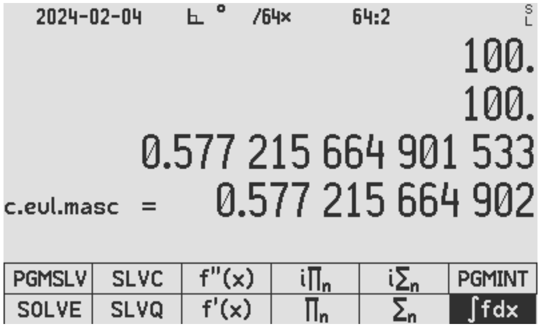
```
C47 Program file export: Export format
version 2, C47 program version 1.
0000: { Prgm #45: 28 bytes / 11 steps }
0001:  LBL 'numer'
0002:    30
0003:    RCL T
0004:    Y^X
0005:    RCL T
0006:    X!
0007:    /
0008:    2
0009:    Y^X
0010:  RTN
0011:  END
```

```
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
```

numerator & 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



Y: Result of the above programmed series

X: Built-in constant γ_{EM}

Subtracting the built-in Euler-Mascheroni constant, the resulting accuracy for p = 2 is shown to be 6.749849059 x 10^{-25}. Raising the arbitrarily chosen p from 2 to 4 (line 8 of [denom] and [numer]) increases the accuracy to 1 x 10^{-34}, see the table below.

p	Terms	Proximity to the Euler Constant
2	30	6.749849059 × 10^{-25}
4	30	-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