GAMMA – Gamma Function and Factorials

Abstract

GAMMA is a program written in 1975 for the HP-25 programmable calculator and compatible models, to accurately evaluate the Gamma function in the interval [1, 2], as well as approximate factorials for real-valued x. Three worked examples are included.

Keywords: *Gamma function, factorial, Stirling's formula, programmable calculator, RPN, HP-25, HP-25C*

1. Introduction

GAMMA is a short *(49 steps) RPN* program that I wrote in 1975 for the *HP-25/25C* programmable calculators (will also run *as-is* or with very minimal modifications in most any *RPN* models) which can accurately compute the value of the *Gamma function*, **$y = \Gamma(x)$**, for x in the interval *[1, 2]*. It uses *Stirling's asymptotic formula*:

$$\Gamma(x+1) \sim \left(\frac{x}{e}\right)^x \sqrt{2\pi x}\,\left(1 + \frac{1}{12x} + \frac{1}{288x^2} - \frac{139}{51840\,x^3}\right)$$

where the given argument **x** is first moved to the interval *[16, 17]* for maximum accuracy, then back to *[1, 2]* by repeated division. This gives results *correct to 9 places over the whole interval*, and even to ~10 places for many arguments. To compute $\Gamma(x)$ for other intervals, repeated multiplication or division may be used. See ***Example 2***.

Also, as a side benefit the program can be easily modified to quickly compute approximate factorials ($x!$) for real **x** (not just integer), $1 \le x \le 69.957$. The accuracy begins at ~4 places for $x = 1$ but steadily improves to ~9 places as x grows bigger. See ***Example 3***.

2. Program Listing

```
01  ENTER↑      15  RCL 0      29  1/x        43  GTO 48 ►
02  1           16  -          30  3          44  STO- 0
03  5           17  e^x        31  y^x        45  R↓
04  +           18  RCL 0      32  RCL 7      46  ÷
05  STO 0       19  1          33  ÷          47  GTO 40 ►
06  ENTER↑      20  2          34  RCL 6    ►48  R↓
07  ln          21  x          35  x          49  R↓
08  x           22  1/x        36  -
09  RCL 0       23  ENTER↑     37  1
10  RCL 5       24  x^2        38  +
11  x           25  2          39  x
12  √x          26  ÷        ►40  RCL 0
13  ln          27  +          41  1
14  +           28  RCL 0      42  x≥y
```

- *49 steps*
- *doesn't alter angular mode or display settings*
- *uses registers R_0, R_5, R_6, R_7*

- *the symbols ► are purely cosmetic, to indicate branching*

3. Usage Instructions

1) First, pre-store the needed constants. This should be done just *once* per session:

[π] 2 [x] [STO 5] 139 [STO 6] 51840 [STO 7]

2) Reset the program counter to the beginning of the program: [f] [PRGM] *(again just once per session)*

3) Key in the argument *1 ≤ x ≤ 2* and run the program: *x* [R/S] *Γ(x)* . Repeat as needed.

4. Examples

The following examples can be useful to check that the program is correctly entered and to understand its usage.

4.1 Example 1

Evaluate *Γ(x)* for *x = 1, 1.1, 1.2, ..., 2* *(you may assume that Steps 1 and 2 above have already been executed)*

[FIX 8] 1 [R/S] *1.00000000* 1.1 [R/S] *0.95135077 ...*

The results are summarized in this table and all are accurate to *9 places*:

1	*1.1*	*1.2*	*1.3*	*1.4*	*1.5*	...	*2*
1	*0.95135077*	*0.91816874*	*0.89747070*	*0.88726382*	*0.88622693*	...	*1.00000001*

4.2 Example 2

Evaluate *Γ(x)* for *x =* π, −π.

Neither π nor −π belong to the *[1, 2]* interval of validity but *π−2 (~1.14)* and *5−π (~1.86)* both do, so we'll first bring the computations to that interval by iteratively using the identity $\Gamma(x+1) = x\,\Gamma(x)$, like this:

$\Gamma(\pi) = \Gamma(\pi-2) * (\pi-2) * (\pi-1)$

[π] 2 [−] [R/S] [π] 2 [−] [x] [π] 1 [−] [x] *2.28803779* (**2.28803779**534..., so we got 9 places)

$\Gamma(-\pi) = \Gamma(5-\pi) / (4-\pi) / (3-\pi) / (2-\pi) / (1-\pi) / (-\pi)$

5 [π] [−] [R/S] 4 [π] [−] [÷] 3 [π] [−] [÷] 2 [π] [−] [÷] 1 [π] [−] [÷] [π] [CHS] [÷] *1.01569715 (ditto)*

4.3 Example 3

The program can be easily modified to quickly compute approximate factorials (*x!*) for real ***x*** (not just integer), in *1 ≤ x ≤ 69.957*. The accuracy begins at ~4 places for *x = 1* but steadily improves to ~9 places as *x* grows bigger. The modification is as follows: in the ***Program Listing*** above,

- replace step *01* ENTER↑ by step *01 GTO 05*
- replace step *40* RCL 0 by step *40 GTO 00*

Once this modification is in place and the program counter has been reset to the beginning of the program by pressing [f] [PRGM] *(just once)*, you can compute the approximated factorial of a real value ***x*** by keying it in and pressing [R/S]. Check it out by computing *x!* for *x = 1, 2, 3, 4* and then for *x = 10, 20.19, 69.*

[FIX 6] 1 [R/S] ***0.9997**11* 2 [R/S] ***1.99998**6* 3 [R/S] ***5.999999*** 4 [R/S] ***24.000003***

[FIX 7] 10 [R/S] ***3628800.0**55* 20.19 [R/S] ***4.3226211 E18*** 69 [R/S] ***1.7112245 E98***

Copyrights