Lanczos approximation

In mathematics, the **Lanczos approximation** is a method for computing the gamma function numerically, published by Cornelius Lanczos in 1964. It is a practical alternative to the more popular Stirling's approximation for calculating the gamma function with fixed precision.

Introduction

The Lanczos approximation consists of the formula

$$\Gamma(z+1) = \sqrt{2\pi}\left(z+g+\tfrac{1}{2}\right)^{z+1/2} e^{-(z+g+1/2)} A_g(z)$$

for the gamma function, with

$$A_g(z) = \frac{1}{2}p_0(g) + p_1(g)\frac{z}{z+1} + p_2(g)\frac{z(z-1)}{(z+1)(z+2)} + \cdots.$$

Here g is a real constant that may be chosen arbitrarily subject to the restriction that $\mathrm{Re}(z+g+\frac{1}{2}) > 0$.[1] The coefficients p, which depend on g, are slightly more difficult to calculate (see below). Although the formula as stated here is only valid for arguments in the right complex half-plane, it can be extended to the entire complex plane by the reflection formula,

$$\Gamma(1-z)\,\Gamma(z) = \frac{\pi}{\sin \pi z}.$$

The series A is convergent, and may be truncated to obtain an approximation with the desired precision. By choosing an appropriate g (typically a small integer), only some 5–10 terms of the series are needed to compute the gamma function with typical single or double floating-point precision. If a fixed g is chosen, the coefficients can be calculated in advance and, thanks to partial fraction decomposition, the sum is recast into the following form:

$$A_g(z) = c_0 + \sum_{k=1}^{N} \frac{c_k}{z+k}$$

Thus computing the gamma function becomes a matter of evaluating only a small number of elementary functions and multiplying by stored constants. The Lanczos approximation was popularized by *Numerical Recipes*, according to which computing the gamma function becomes "not much more difficult than other built-in functions that we take for granted, such as sin x or e^x." The method is also implemented in the GNU Scientific Library, Boost, CPython and musl.

Coefficients

The coefficients are given by

$$p_k(g) = \frac{\sqrt{2}}{\pi} \sum_{\ell=0}^{k} C_{2k+1,\,2\ell+1} \left(\ell - \tfrac{1}{2}\right)! \left(\ell + g + \tfrac{1}{2}\right)^{-(\ell+1/2)} e^{\ell+g+1/2}$$

where $C_{n,m}$ represents the (*n*, *m*)th element of the matrix of coefficients for the Chebyshev polynomials, which can be calculated recursively from these identities:

$$\begin{aligned}
C_{1,\,1} &= 1 \\
C_{2,\,2} &= 1 \\
C_{n+1,\,1} &= -\,C_{n-1,\,1} & &\text{for } n = 2,3,4\,\dots \\
C_{n+1,\,n+1} &= 2\,C_{n,\,n} & &\text{for } n = 2,3,4\,\dots \\
C_{n+1,\,m+1} &= 2\,C_{n,\,m} - C_{n-1,\,m+1} & &\text{for } n > m = 1,2,3\,\dots
\end{aligned}$$

Godfrey (2001) describes how to obtain the coefficients and also the value of the truncated series *A* as a matrix product.[2]

Derivation

Lanczos derived the formula from Leonhard Euler's integral

$$\Gamma(z+1) = \int_0^\infty t^z\, e^{-t}\, dt,$$

performing a sequence of basic manipulations to obtain

$$\Gamma(z+1) = (z+g+1)^{z+1} e^{-(z+g+1)} \int_0^e \Big(v(1-\log v)\Big)^z v^g\, dv,$$

and deriving a series for the integral.

Simple implementation

The following implementation in the Python programming language works for complex arguments and typically gives 13 correct decimal places. Note that omitting the smallest coefficients (in pursuit of speed, for example) gives totally inaccurate results; the coefficients must be recomputed from scratch for an expansion with fewer terms.

```python
from cmath import sin, sqrt, pi, exp


"""
The coefficients used in the code are for when g = 7 and n = 9
Here are some other samples

g = 5
n = 5
p = [
    1.0000018972739440364,
    76.18008222264213732 2,
    -86.505092037054859197,
    24.012898581922685900,
    -1.2296028490285820771
]

g = 5
n = 7
p = [
    1.0000000001900148240,
    76.180091729471463483,
    -86.505320329416767652,
    24.014098240830910490,
    -1.2317395724501553875,
    0.0012086509738661785061,
    -5.3952393849531283785e-6
]

g = 8
n = 12
p = [
    0.9999999999999999298,
    1975.3739023578852322,
    -4397.3823927922428918,
    3462.6328459862717019,
    -1156.9851431631167820,
    154.53815050252775060,
    -6.2536716123689161798,
    0.034642762454736807441,
    -7.4776171974442977377e-7,
    6.3041253821852264261e-8,
    -2.7405717035683877489e-8,
    4.0486948817567609101e-9
]
"""

g = 7
n = 9
p = [
    0.99999999999980993,
```

```
    676.5203681218851,
    -1259.1392167224028,
    771.32342877765313,
    -176.61502916214059,
    12.507343278686905,
    -0.13857109526572012,
    9.9843695780195716e-6,
    1.5056327351493116e-7
]

EPSILON = 1e-07
def drop_imag(z):
    if abs(z.imag) <= EPSILON:
        z = z.real
    return z

def gamma(z):
    z = complex(z)
    if z.real < 0.5:
        y = pi / (sin(pi * z) * gamma(1 - z))  # Reflection formula
    else:
        z -= 1
        x = p[0]
        for i in range(1, len(p)):
            x += p[i] / (z + i)
        t = z + g + 0.5
        y = sqrt(2 * pi) * t ** (z + 0.5) * exp(-t) * x
    return drop_imag(y)
"""
The above use of the reflection (thus the if-else structure) is
necessary, even though
it may look strange, as it allows to extend the approximation to
values of z where
Re(z) < 0.5, where the Lanczos method is not valid.
"""

print(gamma(1))
print(gamma(5))
print(gamma(0.5))
```

See also

- Stirling's approximation
- Spouge's approximation

References

1. Pugh, Glendon (2004). *An analysis of the Lanczos Gamma approximation* (https://web.viu.ca/pughg/phdThesis/phdThesis.pdf#110) (PDF) (Ph.D.).
2. Godfrey, Paul (2001). "Lanczos implementation of the gamma function" (http://www.numericana.com/answer/info/godfrey.htm). *Numericana*.

- Godfrey, Paul (2001). "Lanczos Implementation of the Gamma Function" (http://www.numericana.com/answer/info/godfrey.htm).
- Lanczos, Cornelius (1964). "A Precision Approximation of the Gamma Function". *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*. **1** (1): 86–96. Bibcode:1964SJNA....1...86L (https://ui.adsabs.harvard.edu/abs/1964SJNA....1...86L). doi:10.1137/0701008 (https://doi.org/10.1137%2F0701008). ISSN 0887-459X (https://search.worldcat.org/issn/0887-459X). JSTOR 2949767 (https://www.jstor.org/stable/2949767).
- Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. (2007), "Section 6.1. Gamma Function" (http://apps.nrbook.com/empanel/index.html?pg=256), *Numerical Recipes: The Art of Scientific Computing* (3rd ed.), New York: Cambridge University Press, ISBN 978-0-521-88068-8
- Pugh, Glendon (2004). *An analysis of the Lanczos Gamma approximation* (http://web.mala.bc.ca/pughg/phdThesis/phdThesis.pdf) (PDF) (PhD thesis).
- Toth, Viktor (2005). "Programmable Calculators: The Lanczos Approximation" (http://www.rskey.org/lanczos.htm).
- Weisstein, Eric W. "Lanczos Approximation" (https://mathworld.wolfram.com/LanczosApproximation.html). *MathWorld*.