1. Introduction
   1. Description

In mathematics, the gamma function is one commonly used extension of the factorial function to complex numbers. The gamma function is defined for all complex numbers except the non-positive integers. For any positive integer n,



Derived by Daniel Bernoulli, for complex numbers with a positive real part, the gamma function is defined via a convergent improper integral:

文本, 信件

描述已自动生成

形状, 直方图

中度可信度描述已自动生成

* 1. Domain

Any complex numbers whose real part are excluding non-positive integers.

C/{n∈Z, n≤0}

* 1. Co-Domain

Real numbers excluding non-positive real numbers.

(0, + infinite)

* 1. Properties



Versions suitable for calculators, the approximation

手机屏幕截图

描述已自动生成

1. Functional Requirements
   1. Assumptions
   2. Requirements

We shall only consider the positive value as input value.

We shall only consider the real number as input value, ignore the imaginary part.

When the function value exceeds the maximum value of a double variable, the function will return NaN.

1. Algorithm
   1. Description

Algorithm No.1:

The first algorithm is based on the Gamma function formula derived by Daniel Bernoulli, using an approximating integrals method called trapezoidal rule. It is used for initial value problems. In calculus, the trapezoidal rule is a technique for approximation the definite integral. This algorithm involves two sub-functions. One is pow function, which given base and exponent as double type arguments, returns base^exponent. And exp function, which given exponent as double type argument, returns e^exponent.

Algorithm No.2:

The second algorithm is based on 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. This algorithm also involves pow function, exp function, and sqrt function which given one double type argument, returns

* 1. Pseudo code

Algorithm No.1:

Function y (argument x){

Double y = s^(x-1)\*e^(-s)

}

Function gamma (argument x){

If x < 0 then throw error message

If x > 170 then return NaN

Let result = 0, intervalGap = 10^-3

While i from 0 to a big number

Then

result = result + 1/2\*intervalGap\*(y(i)+y(i-intervalGap))

i = i + intervalGap

End loop

Return result

}

Algorithm No.2:

Declare the arrayList p[] as a coefficients, and a constant EPSILON

Function gamma (argument x){

If x < 0.5 then return (PI / (sin(PI \*z)\*gamma(1-z)))

Else then

x-=1

let z = 0.99999999999980993

for (i, pval) in arraryList p

then z += pval / (x+i+1)

end

let t = x+length(p)-0.5

let m = sqrt(2\*PI)\*pow(t,(x+0.5))\*exp(-t)\*z

Return m

}

3.2 Advantages and Disadvantages

1. References