NAG Toolbox

nag_stat_pdf_gamma (g01kf)

1 Purpose

nag_stat_pdf_gamma (g01kf) returns the value of the probability density function (PDF) for the gamma distribution with shape argument α and scale argument β at a point x.

2 Syntax

```
[result, ifail] = nag_stat_pdf_gamma(x, a, b)
[result, ifail] = g01kf(x, a, b)
```

3 Description

The gamma distribution has PDF

$$f(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta} \quad \text{if } x \geq 0; \quad \alpha, \beta > 0$$

$$f(x) = 0 \quad \text{otherwise.}$$

If $0.01 \leq x, \alpha, \beta \leq 100$ then an algorithm based directly on the gamma distribution's PDF is used. For values outside this range, the function is calculated via the Poisson distribution's PDF as described in Loader (2000) (see Section 9).

4 References

Loader C (2000) Fast and accurate computation of binomial probabilities **(not yet published)**

5 Parameters

5.1 Compulsory Input Parameters

1: **x** – REAL (KIND=nag_wp)

x, the value at which the PDF is to be evaluated.

2: **a** – REAL (KIND=nag_wp)

α, the shape argument of the gamma distribution.

Constraint: **a** > 0.0.

3: **b** – REAL (KIND=nag_wp)

β, the scale argument of the gamma distribution.

Constraints:

b > 0.0;
$\frac{\mathbf{x}}{\mathbf{b}} < \frac{1}{\text{x02am()}}$.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **result**

The result of the function.

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

If **ifail** $\neq 0$, then nag_stat_pdf_gamma (g01kf) returns 0.0.

ifail $= 1$

Constraint: **a** > 0.0.

ifail $= 2$

Constraint: **b** > 0.0.

ifail $= 3$

Computation abandoned owing to overflow due to extreme parameter values.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

Due to the lack of a stable link to Loader (2000) paper, we give a brief overview of the method, as applied to the Poisson distribution. The Poisson distribution has a continuous mass function given by,

$$p(x;\lambda) = \frac{\lambda^x}{x!}e^{-\lambda}. \tag{1}$$

The usual way of computing this quantity would be to take the logarithm and calculate,

$$\log(x;\lambda) = x\log\lambda - \log(x!) - \lambda.$$

For large x and λ, $x\log\lambda$ and $\log(x!)$ are very large, of the same order of magnitude and when calculated have rounding errors. The subtraction of these two terms can therefore result in a number, many orders of magnitude smaller and hence we lose accuracy due to subtraction errors. For example for $x = 2 \times 10^6$ and $\lambda = 2 \times 10^6$, $\log(x!) \approx 2.7 \times 10^7$ and $\log(p(x;\lambda)) = -8.17326744645834$. But calculated with the method shown later we have $\log(p(x;\lambda)) = -8.1732674441334492$. The difference between these two results suggests a loss of about 7 significant figures of precision.

Loader introduces an alternative way of expressing (1) based on the saddle point expansion,

$$\log(p(x;\lambda)) = \log(p(x;x)) - D(x;\lambda), \tag{2}$$

where $D(x;\lambda)$, the deviance for the Poisson distribution is given by,

$$\begin{aligned} D(x;\lambda) &= \log(p(x;x)) - \log(p(x;\lambda)), \\ &= \lambda D_0\left(\tfrac{x}{\lambda}\right), \end{aligned} \tag{3}$$

and

$$D_0(\epsilon) = \epsilon\log\epsilon + 1 - \epsilon.$$

For ϵ close to 1, $D_0(\epsilon)$ can be evaluated through the series expansion

$$\lambda D_0\left(\frac{x}{\lambda}\right) = \frac{(x-\lambda)^2}{x+\lambda} + 2x\sum_{j=1}^{\infty}\frac{v^{2j+1}}{2j+1}, \qquad \text{where } v = \frac{x-\lambda}{x+\lambda},$$

otherwise $D_0(\epsilon)$ can be evaluated directly. In addition, Loader suggests evaluating $\log(x!)$ using the Stirling–De Moivre series,

$$\log(x!) = \frac{1}{2}\log(2\pi x) + x\log(x) - x + \delta(x), \tag{4}$$

where the error $\delta(x)$ is given by

$$\delta(x) = \frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5} + \mathcal{O}\left(x^{-7}\right).$$

Finally $\log(p(x;\lambda))$ can be evaluated by combining equations (1)–(4) to get,

$$p(x;\lambda) = \frac{1}{\sqrt{2\pi x}} e^{-\delta(x) - \lambda D_0(x/\lambda)}.$$

9 Example

This example prints the value of the gamma distribution PDF at six different points **x** with differing **a** and **b**.

9.1 Program Text

```
     function g01kf_example

fprintf('g01kf example results\n\n');

x      = [0.1, 3, 6,  4,   9,  16];
a      = [3,  10, 5, 10,   9,   3.5];
b      = [2,  11, 1,  0.1, 0.5, 2.5];
result = x;
fprintf('\n      x            a            b          pdf\n');

for i=1:numel(x)
 [result(i), ifail] = g01kf( ...
      x(i), a(i), b(i));
end

fprintf('%12.4e %12.4e %12.4e %12.4e\n', [x; a; b; result]);

g01kf_plot;

function g01kf_plot

  fig1 = figure;
  hold on;
  a  = [2, 9];
  b  = [2, 0.5];
  x  = [0:0.1:10];
  alpha = '\alpha';
```

```
  beta  = '\beta';
  for i=1:2
    for j=1:numel(x)
      [y{i}(j), ifail] = g01kf( ...
x(j), a(i), b(i));
    end
    plot(x,y{i});
    l{i} = sprintf('%s = %3.1f, %s = %3.1f', alpha, a(i), beta, b(i));
  end
  legend(l);
  xlabel('x');
  title('Gamma Distributions');
  hold off;
  % print(fig1,'-dpng','-r75','g01kf_fig1.png');
  % print(fig1,'-deps','-r75','g01kf_fig1.eps');
```

9.2 Program Results

```
g01kf example results

     x             a             b            pdf
 1.0000e-01   3.0000e+00   2.0000e+00   5.9452e-04
 3.0000e+00   1.0000e+01   1.1000e+01   1.5921e-12
 6.0000e+00   5.0000e+00   1.0000e+00   1.3385e-01
 4.0000e+00   1.0000e+01   1.0000e-01   3.0690e-08
 9.0000e+00   9.0000e+00   5.0000e-01   8.3251e-03
 1.6000e+01   3.5000e+00   2.5000e+00   2.0723e-02
```