

[Help](#)

```
#include <stdio.h>
#include <stdlib.h>
#include <
href../../../../common/math/cdo/cdo_math_h_src.pdfmath.h>

#include "pnl/pnl_cdf.h"
#include "
href../../../../common/math/cdo/cdo_math_h_src.pdfcdo_math.h"
#include "pnl/pnl_specfun.h"

double nig_generic_density(double x, double alpha, double beta, double gamma, double delta)
{
    double f_x = sqrt(delta * delta + (x - mu) * (x - mu));
    return ((delta * alpha * exp(delta * gamma + beta * (x - mu)) *
            pnl_bessel_k(1., alpha * f_x)) / (M_PI * f_x));
}

double ig_generic_density(double y, double alpha, double beta)
{
    double z = alpha - beta * y;

    if (y <= 0) return (0.);
    return (M_1_SQRT2PI * (alpha / sqrt(beta)) * pow(y, -1.5) * exp(- z * z / (2. * beta)));
}

double nig_generic_cdf(double x, double alpha, double beta, double gamma, double delta)
{
    double y;
    double z;
    double t;
    double h;
    double s1;
    double s2;

    s1 = 0;
    h = 4. / 100.;
    for (y = MINDOUBLE; y < 4.; y += h)
    {
```

```

        z = (x - (mu + beta * (y + 0.5 * h))) / sqrt(y + 0.5 * h);
        s1 += cdf_nor(z) * ig_generic_density(y + 0.5 * h, delta * gamma, gamma *
    }
    s1 *= h;
    s2 = 0;
    h = exp(-4.) / 20.;
    for (t = MINDOUBLE; t < exp(-4.); t += h)
    {
        y = -log(t + 0.5 * h);
        z = (x - (mu + beta * y)) / sqrt(y);
        s2 += cdf_nor(z) * ig_generic_density(y, delta * gamma, gamma * gamma) * (
    }
    s2 *= h;

    return (s1 + s2);
};

```