nag_bivariate_normal_dist (g01hac) returns the lower tail probability for the bivariate Normal distribution.

The lower tail probability is defined by:

\[
P(X \leq x, Y \leq y : \rho) = \frac{1}{2\pi\sqrt{1 - \rho^2}} \int_{-\infty}^{y} \int_{-\infty}^{x} \exp\left(-\frac{(X^2 - 2\rho XY + Y^2)}{2(1 - \rho^2)}\right) dXdY.
\]

For a more detailed description of the bivariate Normal distribution and its properties see Abramowitz and Stegun (1972) and Kendall and Stuart (1969). The method used is described by Genz (2004).

### Arguments

1. **x** – double
   
   *Input*
   
   *On entry:* x, the first argument for which the bivariate Normal distribution function is to be evaluated.

2. **y** – double
   
   *Input*
   
   *On entry:* y, the second argument for which the bivariate Normal distribution function is to be evaluated.

3. **rho** – double
   
   *Input*
   
   *On entry:* ρ, the correlation coefficient.
   
   *Constraint:* \(-1.0 \leq \rho \leq 1.0\).
6 Error Indicators and Warnings

On any of the error conditions listed below nag_bivariate_normal_dist (g01hac) returns 0.0.

**NE_ALLOC_FAIL**
Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

**NE_INTERNAL_ERROR**
An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.
An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

**NE_NO_LICENCE**
Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

**NE_REAL_ARG_GT**
On entry, \( \rho = \langle \text{value} \rangle \).
Constraint: \( \rho \leq 1.0 \).

**NE_REAL_ARG_LT**
On entry, \( \rho = \langle \text{value} \rangle \).
Constraint: \( \rho \geq -1.0 \).

7 Accuracy

Accuracy of the hybrid algorithm implemented here is discussed in Genz (2004). This algorithm should give a maximum absolute error of less than \( 5 \times 10^{-16} \).

8 Parallelism and Performance

Not applicable.

9 Further Comments

The probabilities for the univariate Normal distribution can be computed using nag_cumul_normal (s15abc) and nag_cumul_normal_complem (s15acc).

10 Example

This example reads values of \( x \) and \( y \) for a bivariate Normal distribution along with the value of \( \rho \) and computes the lower tail probabilities.
10.1 Program Text

```c
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nag01.h>

int main(void)
{
    Integer exit_status = 0;
    double prob, rho, x, y;
    NagError fail;

    INIT_FAIL(fail);

    /* Skip heading in data file */
    #ifdef _WIN32
        scanf_s("%*[\n]");
    #else
        scanf("%*[\n]");
    #endif
    printf("nag_bivariate_normal_dist (g01hac) Example Program Results\n");
    printf(" x y rho prob\n\n");
    #ifdef _WIN32
        while (scanf_s("%lf %lf %lf", &x, &y, &rho) != EOF)
    #else
        while (scanf("%lf %lf %lf", &x, &y, &rho) != EOF)
    #endif
    {
        /* nag_bivariate_normal_dist (g01hac).
           * Probability for the bivariate Normal distribution
           */
        prob = nag_bivariate_normal_dist(x, y, rho, &fail);
        if (fail.code != NE_NOERROR)
        {
            printf("Error from nag_bivariate_normal_dist (g01hac).\n%s\n", fail.message);
            exit_status = 1;
            goto END;
        }
        printf("%8.3f%8.3f%8.4f\n", x, y, rho, prob);
    }

END:
    return exit_status;
}
```

10.2 Program Data

```
nag_bivariate_normal_dist (g01hac) Example Program Data
1.7 23.1 0.0
0.0 0.0 0.1
3.3 11.1 0.54
9.1 9.1 0.17
```

Mark 25 g01hac.3
### 10.3 Program Results

**nag_bivariate_normal_dist (g01hac) Example Program Results**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>rho</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.700</td>
<td>23.100</td>
<td>0.000</td>
<td>0.9554</td>
</tr>
<tr>
<td>0.000</td>
<td>0.000</td>
<td>0.100</td>
<td>0.2659</td>
</tr>
<tr>
<td>3.300</td>
<td>11.100</td>
<td>0.540</td>
<td>0.9995</td>
</tr>
<tr>
<td>9.100</td>
<td>9.100</td>
<td>0.170</td>
<td>1.0000</td>
</tr>
</tbody>
</table>