

## NAG Library Function Document

### nag\_init\_vavilov (g01zuc)

#### 1 Purpose

nag\_init\_vavilov (g01zuc) is used to initialize functions nag\_prob\_vavilov (g01euc) and nag\_prob\_density\_vavilov (g01muc).

It is intended to be used before a call to nag\_prob\_vavilov (g01euc) or nag\_prob\_density\_vavilov (g01muc).

#### 2 Specification

```
#include <nag.h>
#include <nagg01.h>

void nag_init_vavilov (double rkappa, double beta2, Integer mode, double *xl,
                     double *xu, double comm_arr[], NagError *fail)
```

#### 3 Description

nag\_init\_vavilov (g01zuc) initializes the array **comm\_arr** for use by nag\_prob\_vavilov (g01euc) or nag\_prob\_density\_vavilov (g01muc) in the evaluation of the Vavilov functions  $\phi_V(\lambda; \kappa, \beta^2)$  and  $\Phi_V(\lambda; \kappa, \beta^2)$  respectively.

Multiple calls to nag\_prob\_vavilov (g01euc) or nag\_prob\_density\_vavilov (g01muc) can be made following a single call to nag\_init\_vavilov (g01zuc), provided that **rkappa** or **beta2** do not change, and that either all calls are to nag\_prob\_vavilov (g01euc) or all calls are to nag\_prob\_density\_vavilov (g01muc). If you wish to call both nag\_prob\_vavilov (g01euc) and nag\_prob\_density\_vavilov (g01muc), then you will need to initialize both separately.

#### 4 References

Schorr B (1974) Programs for the Landau and the Vavilov distributions and the corresponding random numbers *Comp. Phys. Comm.* **7** 215–224

#### 5 Arguments

- 1: **rkappa** – double *Input*  
*On entry:* the argument  $\kappa$  of the function.  
*Constraint:*  $0.01 \leq \mathbf{rkappa} \leq 10.0$ .
- 2: **beta2** – double *Input*  
*On entry:* the argument  $\beta^2$  of the function.  
*Constraint:*  $0.0 \leq \mathbf{beta2} \leq 1.0$ .
- 3: **mode** – Integer *Input*  
*On entry:* if **mode** = 0, then nag\_prob\_density\_vavilov (g01muc) is to be called after the call to nag\_init\_vavilov (g01zuc). Otherwise, nag\_prob\_vavilov (g01euc) is to be called.

- 4: **x<sub>l</sub>** – double \* *Output*  
*On exit:*  $x_l$ , a threshold value below which  $\phi_V(\lambda; \kappa, \beta^2)$  will be set to zero by nag\_prob\_density\_vavilov (g01muc) and  $\Phi_V(\lambda; \kappa, \beta^2)$  will be set to zero by nag\_prob\_vavilov (g01euc) if  $\lambda < x_l$ .
- 5: **x<sub>u</sub>** – double \* *Output*  
*On exit:*  $x_u$ , a threshold value above which  $\phi_V(\lambda; \kappa, \beta^2)$  will be set to zero by nag\_prob\_density\_vavilov (g01muc) and  $\Phi_V(\lambda; \kappa, \beta^2)$  will be set to unity by nag\_prob\_vavilov (g01euc) if  $\lambda > x_u$ .
- 6: **comm\_arr[322]** – double *Communication Array*  
*On exit:* this argument should be passed unchanged to nag\_prob\_vavilov (g01euc) or nag\_prob\_density\_vavilov (g01muc).
- 7: **fail** – NagError \* *Input/Output*  
 The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

## 6 Error Indicators and Warnings

### NE\_ALLOC\_FAIL

Dynamic memory allocation failed.

See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

### NE\_BAD\_PARAM

On entry, argument *<value>* had an illegal value.

### 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 How to Use the NAG Library and its Documentation 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 How to Use the NAG Library and its Documentation for further information.

### NE\_REAL

On entry, **beta2** = *<value>*.

Constraint: **beta2** ≤ 1.0.

On entry, **beta2** = *<value>*.

Constraint: **beta2** ≥ 0.0.

On entry, **rkappa** = *<value>*.

Constraint: **rkappa** ≤ 10.0.

On entry, **rkappa** = *<value>*.

Constraint: **rkappa** ≥ 0.01.

## **7 Accuracy**

At least five significant digits are usually correct.

## **8 Parallelism and Performance**

nag\_init\_vavilov (g01zuc) is not threaded in any implementation.

## **9 Further Comments**

None.

## **10 Example**

See Section 10 in nag\_prob\_density\_vavilov (g01muc) and nag\_prob\_vavilov (g01euc).

---