NAG Library Function Document

nag_quasi_init (g05ylc)

1 Purpose

nag_quasi_init (g05ylc) initializes a quasi-random generator prior to calling nag_quasi_rand_normal (g05yjc), nag_quasi_rand_lognormal (g05ykc) or nag_quasi_rand_uniform (g05ymc).

2 Specification

```
#include <nag.h>
#include <nagg05.h>
void nag_quasi_init (Nag_QuasiRandom_Sequence genid, Integer idim,
     Integer iref[], Integer liref, Integer iskip, NagError *fail)
```

3 Description

nag_quasi_init (g05ylc) selects a quasi-random number generator through the input value of **genid** and initializes the **iref** communication array for use by the functions nag_quasi_rand_normal (g05yjc), nag_quasi_rand_lognormal (g05ykc) or nag_quasi_rand_uniform (g05ymc).

One of three types of quasi-random generator may be chosen, allowing the low-discrepancy sequences proposed by Sobol, Faure or Niederreiter to be generated.

Two sets of Sobol sequences are supplied, the first, is based on the work of Joe and Kuo (2008). The second, referred to in the documentation as "Sobol (A659)", is based on Algorithm 659 of Bratley and Fox (1988) with the extension to 1111 dimensions proposed by Joe and Kuo (2003). Both sets of Sobol sequences should satisfy the so-called Property A, up to 1111 dimensions, but the first set should have better two-dimensional projections than those produced using Algorithm 659.

4 References

Bratley P and Fox B L (1988) Algorithm 659: implementing Sobol's quasirandom sequence generator *ACM Trans. Math. Software* **14(1)** 88–100

Fox B L (1986) Algorithm 647: implementation and relative efficiency of quasirandom sequence generators *ACM Trans. Math. Software* **12(4)** 362–376

Joe S and Kuo F Y (2003) Remark on Algorithm 659: implementing Sobol's quasirandom sequence generator *ACM Trans. Math. Software (TOMS)* **29** 49–57

Joe S and Kuo F Y (2008) Constructing Sobol sequences with better two-dimensional projections *SIAM J. Sci. Comput.* **30** 2635–2654

5 Arguments

1: **genid** – Nag_QuasiRandom_Sequence *Input*

On entry: must identify the quasi-random generator to use.

genid = Nag_QuasiRandom_Sobol
Sobol generator.

genid = Nag_QuasiRandom_SobolA659
Sobol (A659) generator.

genid = Nag_QuasiRandom_Nied
Niederreiter generator.

genid = Nag_QuasiRandom_Faure
Faure generator.

Constraint: **genid** = Nag_QuasiRandom_Sobol, Nag_QuasiRandom_SobolA659, Nag_QuasiRandom_Nied or Nag_QuasiRandom_Faure.

2: **idim** – Integer *Input*

On entry: the number of dimensions required.

Constraints:

if **genid** = Nag_QuasiRandom_Sobol, $1 \leq$ **idim** ≤ 10000;
if **genid** = Nag_QuasiRandom_SobolA659, $1 \leq$ **idim** ≤ 1111;
if **genid** = Nag_QuasiRandom_Nied, $1 \leq$ **idim** ≤ 318;
if **genid** = Nag_QuasiRandom_Faure, $1 \leq$ **idim** ≤ 40.

3: **iref**[**liref**] – Integer *Communication Array*

On exit: contains initialization information for use by the generator functions nag_quasi_r and_normal (g05yjc), nag_quasi_rand_lognormal (g05ykc) and nag_quasi_rand_uniform (g05ymc). **iref** must not be altered in any way between initialization and calls of the generator functions.

4: **liref** – Integer *Input*

On entry: the dimension of the array **iref**.

Constraints:

if **genid** = Nag_QuasiRandom_Sobol, Nag_QuasiRandom_SobolA659 or Nag_QuasiRandom_Nied, **liref** $\geq 32 \times$ **idim** $+ 7$;
if **genid** = Nag_QuasiRandom_Faure, **liref** ≥ 407.

5: **iskip** – Integer *Input*

On entry: the number of terms of the sequence to skip on initialization for the Sobol and Niederreiter generators. If **genid** = Nag_QuasiRandom_Faure, **iskip** is ignored.

Constraint: if **genid** = Nag_QuasiRandom_Sobol, Nag_QuasiRandom_SobolA659 or Nag_QuasiRandom_Nied, $0 \leq$ **iskip** $\leq 2^{30}$.

6: **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_INT

On entry, **idim** = ⟨*value*⟩.
Constraint: $1 \leq$ **idim** $\leq$ ⟨*value*⟩.

On entry, **iskip** < 0 or **iskip** is too large: **iskip** = ⟨*value*⟩, maximum value is ⟨*value*⟩.

On entry, **liref** is too small: **liref** = ⟨*value*⟩, minimum length is ⟨*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.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_quasi_init (g05ylc) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the x06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users' Notefor your implementation for any additional implementation-specific information.

9 Further Comments

The primitive polynomials and direction numbers used for the Sobol generator (**genid** = Nag_QuasiRandom_Sobol) were calculated by Joe and Kuo (2008) using the search critera $D^{(6)}$.

10 Example

See Section 10 in nag_quasi_rand_uniform (g05ymc).