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

```c
#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


5 Arguments

1:  

```c
Nag_QuasiRandom_Sequence genid
```

*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.
\[
\text{genid} = \text{Nag\_QuasiRandom\_Faure}
\]
Faure generator.

*Constraint*: \text{genid} = \text{Nag\_QuasiRandom\_Sobol}, \text{Nag\_QuasiRandom\_SobolA659}, \text{Nag\_QuasiRandom\_Nied} or \text{Nag\_QuasiRandom\_Faure}.

2: \text{idim} \text{ – Integer} \quad \text{Input}

*On entry*: the number of dimensions required.

*Constraints*:
- if \text{genid} = \text{Nag\_QuasiRandom\_Sobol}, \(1 \leq \text{idim} \leq 10000\);
- if \text{genid} = \text{Nag\_QuasiRandom\_SobolA659}, \(1 \leq \text{idim} \leq 1111\);
- if \text{genid} = \text{Nag\_QuasiRandom\_Nied}, \(1 \leq \text{idim} \leq 318\);
- if \text{genid} = \text{Nag\_QuasiRandom\_Faure}, \(1 \leq \text{idim} \leq 40\).

3: \text{iref[iref]} \text{ – Integer} \quad \text{Communication Array}

*On exit*: contains initialization information for use by the generator functions \text{nag\_quasi\_rand\_normal} (g05yjc), \text{nag\_quasi\_rand\_lognormal} (g05ykc) and \text{nag\_quasi\_rand\_uniform} (g05ymc). \text{iref} must not be altered in any way between initialization and calls of the generator functions.

4: \text{liref} \text{ – Integer} \quad \text{Input}

*On entry*: the dimension of the array \text{iref}.

*Constraints*:
- if \text{genid} = \text{Nag\_QuasiRandom\_Sobol}, \text{Nag\_QuasiRandom\_SobolA659} or \text{Nag\_QuasiRandom\_Nied}, \text{liref} \geq 32 \times \text{idim} + 7;
- if \text{genid} = \text{Nag\_QuasiRandom\_Faure}, \text{liref} \geq 407.

5: \text{iskip} \text{ – Integer} \quad \text{Input}

*On entry*: the number of terms of the sequence to skip on initialization for the Sobol and Niederreiter generators. If \text{genid} = \text{Nag\_QuasiRandom\_Faure}, \text{iskip} is ignored.

*Constraint*: if \text{genid} = \text{Nag\_QuasiRandom\_Sobol}, \text{Nag\_QuasiRandom\_SobolA659} or \text{Nag\_QuasiRandom\_Nied}, \(0 \leq \text{iskip} \leq 2^{30}\).

6: \text{fail} \quad \text{NagError*} \quad \text{Input/Output}

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 \quad \text{Error Indicators and Warnings}

\text{NE\_ALLOC\_FAIL}
Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

\text{NE\_BAD\_PARAM}
On entry, argument \(\langle\text{value}\rangle\) had an illegal value.

\text{NE\_INT}
On entry, \text{idim} = \(\langle\text{value}\rangle\).
Constraint: \(1 \leq \text{idim} \leq \langle\text{value}\rangle\).
On entry, \text{iskip} < 0 or \text{iskip} is too large: \text{iskip} = \(\langle\text{value}\rangle\), maximum value is \(\langle\text{value}\rangle\).
On entry, \text{liref} is too small: \text{liref} = \(\langle\text{value}\rangle\), minimum length is \(\langle\text{value}\rangle\).
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.

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’ Note for 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 criteria $D^{(6)}$.

10 Example
See Section 10 in nag_quasi_rand_uniform (g05ymc).