NAG Library Function Document

nag_quad_md_numth_coeff_prime (d01gyc)

1 Purpose

nag_quad_md_numth_coeff_prime (d01gyc) calculates the optimal coefficients for use by nag_quad_md_numth_vec (d01gdc), for prime numbers of points.

2 Specification

```
#include <nag.h>
#include <nagd01.h>
```

```
void nag_quad_md_numth_coeff_prime (Integer ndim, Integer npts, double vk[],
     NagError *fail)
```

3 Description

The Korobov (1963) procedure for calculating the optimal coefficients $a_1, a_2, \ldots, a_n$ for p-point integration over the n-cube $[0,1]^n$ imposes the constraint that

$$a_1 = 1 \quad \text{and} \quad a_i = a^{i-1} (\bmod p), \quad i = 1, 2, \ldots, n \tag{1}$$

where p is a prime number and a is an adjustable argument. This argument is computed to minimize the error in the integral

$$3^n \int_0^1 dx_1 \cdots \int_0^1 dx_n \prod_{i=1}^{n} (1 - 2x_i)^2, \tag{2}$$

when computed using the number theoretic rule, and the resulting coefficients can be shown to fit the Korobov definition of optimality.

The computation for large values of p is extremely time consuming (the number of elementary operations varying as p^2) and there is a practical upper limit to the number of points that can be used. Function nag_quad_md_numth_coeff_2prime (d01gzc) is computationally more economical in this respect but the associated error is likely to be larger.

4 References

Korobov N M (1963) *Number Theoretic Methods in Approximate Analysis* Fizmatgiz, Moscow

5 Arguments

1: **ndim** – Integer *Input*

On entry: n, the number of dimensions of the integral.

Constraint: **ndim** ≥ 1.

2: **npts** – Integer *Input*

On entry: p, the number of points to be used.

Constraint: **npts** must be a prime number ≥ 5.

3: **vk**[**ndim**] – double *Output*

On exit: the n optimal coefficients.

4: **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_ACCURACY

The ***machine precision*** is insufficient to perform the computation exactly. Try reducing **npts**: **npts** $= \langle value \rangle$.

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

NE_INT

On entry, **ndim** $= \langle value \rangle$.
Constraint: **ndim** ≥ 1.

On entry, **npts** $= \langle value \rangle$.
Constraint: **npts** must be a prime number.

On entry, **npts** $= \langle value \rangle$.
Constraint: **npts** ≥ 5.

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 2.7.6 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 2.7.5 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The optimal coefficients are returned as exact integers (though stored in a double array).

8 Parallelism and Performance

nag_quad_md_numth_coeff_prime (d01gyc) is not threaded in any implementation.

9 Further Comments

The time taken is approximately proportional to p^2 (see Section 3).

10 Example

This example calculates the Korobov optimal coefficients where the number of dimensions is 4 and the number of points is 631.

10.1 Program Text

```
/* nag_quad_md_numth_coeff_prime (d01gyc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagd01.h>

int main(void)
{
  Integer exit_status = 0;
  Integer i, ndim, npts;
  double *vk = 0;
  NagError fail;

  INIT_FAIL(fail);

  printf("nag_quad_md_numth_coeff_prime (d01gyc) Example Program Results\n");
  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif
#ifdef _WIN32
  scanf_s("%" NAG_IFMT "", &ndim);
#else
  scanf("%" NAG_IFMT "", &ndim);
#endif
#ifdef _WIN32
  scanf_s("%" NAG_IFMT "%*[^\n] ", &npts);
#else
  scanf("%" NAG_IFMT "%*[^\n] ", &npts);
#endif

  if (!(vk = NAG_ALLOC(ndim, double)))
  {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }

  /* nag_quad_md_numth_coeff_prime (d01gyc).
   * Korobov optimal coefficients for use in nag_quad_md_numth_vec (d01gdc),
   * when number of points is prime.
   */
  nag_quad_md_numth_coeff_prime(ndim, npts, vk, &fail);
  if (fail.code != NE_NOERROR) {
    printf("Error from nag_quad_md_numth_coeff_prime (d01gyc).\n%s\n",
           fail.message);
    exit_status = 1;
    goto END;
  }

  printf("\nndim = %3" NAG_IFMT " npts = %6" NAG_IFMT "\n", ndim, npts);
  printf("\nCoefficients =");
  for (i = 0; i < ndim; i++)
    printf("%4.0f ", vk[i]);
  printf("\n");
```

```
END:
  NAG_FREE(vk);

  return exit_status;
}
```

10.2 Program Data

None.

10.3 Program Results

```
nag_quad_md_numth_coeff_prime (d01gyc) Example Program Results

ndim =   4 npts =    631

Coefficients =   1  198   82  461
```