

## NAG Library Function Document

### nag\_quad\_md\_numth\_coeff\_2prime (d01gzc)

#### 1 Purpose

nag\_quad\_md\_numth\_coeff\_2prime (d01gzc) calculates the optimal coefficients for use by nag\_quad\_md\_numth\_vec (d01gdc), when the number of points is the product of two primes.

#### 2 Specification

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

void nag_quad_md_numth_coeff_2prime (Integer ndim, Integer np1, Integer np2,
    double vk[], NagError *fail)
```

#### 3 Description

Korobov (1963) gives a procedure for calculating optimal coefficients for  $p$ -point integration over the  $n$ -cube  $[0, 1]^n$ , when the number of points is

$$p = p_1 p_2 \tag{1}$$

where  $p_1$  and  $p_2$  are distinct prime numbers.

The advantage of this procedure is that if  $p_1$  is chosen to be the nearest prime integer to  $p_2^2$ , then the number of elementary operations required to compute the rule is of the order of  $p^{4/3}$  which grows less rapidly than the number of operations required by nag\_quad\_md\_numth\_coeff\_prime (d01gyc). The associated error is likely to be larger although it may be the only practical alternative for high values of  $p$ .

#### 4 References

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

#### 5 Arguments

- |    |  |        |
|----|--|--------|
| 1: | <b>ndim</b> – Integer<br><i>On entry:</i> $n$ , the number of dimensions of the integral.<br><i>Constraint:</i> <b>ndim</b> $\geq 1$ .   | Input  |
| 2: | <b>np1</b> – Integer<br><i>On entry:</i> the larger prime factor $p_1$ of the number of points in the integration rule.<br><i>Constraint:</i> <b>np1</b> must be a prime number $\geq 5$ .   | Input  |
| 3: | <b>np2</b> – Integer<br><i>On entry:</i> the smaller prime factor $p_2$ of the number of points in the integration rule. For maximum efficiency, $p_2^2$ should be close to $p_1$ .<br><i>Constraint:</i> <b>np2</b> must be a prime number such that <b>np1</b> $>$ <b>np2</b> $\geq 2$ . | Input  |
| 4: | <b>vk[ndim]</b> – double<br><i>On exit:</i> the $n$ optimal coefficients.  | Output |

5: **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 **np1** or **np2**: **np1** =  $\langle value \rangle$  and **np2** =  $\langle value \rangle$ .

### 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**  $\geq 1$ .

On entry, **np1** =  $\langle value \rangle$ .

Constraint: **np1** must be a prime number.

On entry, **np1** =  $\langle value \rangle$ .

Constraint: **np1**  $\geq 5$ .

On entry, **np2** =  $\langle value \rangle$ .

Constraint: **np2** must be a prime number.

On entry, **np2** =  $\langle value \rangle$ .

Constraint: **np2**  $\geq 2$ .

### NE\_INT\_2

On entry, **np1**  $\times$  **np2** exceeds largest machine integer. **np1** =  $\langle value \rangle$  and **np2** =  $\langle value \rangle$ .

On entry, **np1** =  $\langle value \rangle$  and **np2** =  $\langle value \rangle$ .

Constraint: **np1**  $>$  **np2**.

### 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\_2prime (d01gzc) is not threaded in any implementation.

## 9 Further Comments

The time taken by nag\_quad\_md\_numth\_coeff\_2prime (d01gzc) grows at least as fast as  $(p_1 p_2)^{4/3}$ . (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 the product of the two prime numbers, 89 and 11.

### 10.1 Program Text

```
/* nag_quad_md_numth_coeff_2prime (d01gzc) 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, np1, np2;
    double *vk = 0;
    NagError fail;

    INIT_FAIL(fail);

    printf("nag_quad_md_numth_coeff_2prime (d01gzc) 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 " %" NAG_IFMT "%*[\n] ", &np1, &np2);
#else
    scanf("%" NAG_IFMT " %" NAG_IFMT "%*[\n] ", &np1, &np2);
#endif

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

    /* nag_quad_md_numth_coeff_2prime (d01gzc).
     * Korobov optimal coefficients for use in d01gdc,
     * when number of points is product of two primes.
     */
}
```

```

    */
    nag_quad_md_numth_coeff_2prime(ndim, np1, np2, vk, &fail);
    if (fail.code != NE_NOERROR) {
        printf("Error from nag_quad_md_numth_coeff_2prime (d01gzc).\n%s\n",
            fail.message);
        exit_status = 1;
        goto END;
    }

    printf("\nndim = %3" NAG_IFMT " np1 = %6" NAG_IFMT " np2 = %6" NAG_IFMT
        "\n", ndim, np1, np2);
    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\_2prime (d01gzc) Example Program Results

ndim =    4 np1 =       89 np2 =       11

Coefficients =    1 102 614 951

---