NAG Library Routine Document

c06saf

1 Purpose

c06saf calculates the multidimensional fast Gauss transform.

2 Specification

```
Subroutine c06saf (d, srcs, n, trgs, m, q, p1, p2, k, hin, lhin, tol, v,    &
                   term, ifail)
Integer, Intent (In)               :: d, n, m, lhin
Integer, Intent (Inout)            :: p1, p2, k, ifail
Real (Kind=nag_wp), Intent (In)    :: srcs(d,n), trgs(d,m), q(n),          &
                                      hin(lhin), tol
Real (Kind=nag_wp), Intent (Out) :: v(m), term(m)
```

3 Description

c06saf calculates the d-dimensional fast Gauss transform (FGT), $\hat{G}(y)$, that approximates the discrete Gauss transform (DGT), $G(y)$, evaluated at a set of target points y_j, for $j = 1, 2, \ldots, m \in R^d$. The DGT is defined as:

$$G(y_j) = \sum_{i=1}^{n} q_i e^{-\left\|y_j - x_i\right\|_2^2 / h_i^2}, \quad j = 1, \ldots, m$$

where x_i, for $i = 1, 2, \ldots, n \in R^d$, are the Gaussian source points, q_i, for $i = 1, 2, \ldots, n \in R^+$, are the source weights and h_i, for $i = 1, 2, \ldots, n \in R^+$, are the source standard deviations (alternatively source scales or source bandwidths).

This subroutine implements the improved FGT algorithm presented in Raykar and Duraiswami (2005). The algorithm clusters the sources into k distinct clusters and then computes two Taylor series approximations per cluster with p_1 and p_2 terms respectively. You must provide p_1, p_2 and k when calling the subroutine. See Section 7 below for a further discussion on accuracy when choosing their values.

The input array **hin** of this routine is designed to allow maximum flexibility in the supply of the standard deviation arguments by reusing, in a cyclic manner, elements of the array when it is less than n elements long. For example, if all Gaussian sources have the same standard deviation then it is only necessary to set **lhin** to 1 and to provide the value of the standard deviation in **hin**(1); the routine will then automatically expand **hin** to be of length n. For further details please see Section 2.6 in the G01 Chapter Introduction.

4 References

Greengard L and Strain J (1991) The Fast Gauss Transform *SIAM J. Sci. Statist. Comput.* **12(1)** 79–94

Raykar V C and Duraiswami R (2005) Improved Fast Gauss Transform With Variable Source Scales *University of Maryland Technical Report CS-TR-4727/UMIACS-TR-2005-34*

5 Arguments

1: **d** – Integer *Input*

On entry: d, the number of dimensions.

Constraint: **d** > 0.

2: **srcs**($\mathbf{d}, \mathbf{n}$) – Real (Kind=nag_wp) array *Input*

On entry: x, the locations of the Gaussian sources.

3: **n** – Integer *Input*

On entry: n, the number of Gaussian sources.

Constraint: $\mathbf{n} > 0$.

4: **trgs**($\mathbf{d}, \mathbf{m}$) – Real (Kind=nag_wp) array *Input*

On entry: y, the locations of the target points at which the FGT will be evaluated.

5: **m** – Integer *Input*

On entry: m, the number of target points.

Constraint: $\mathbf{m} > 0$.

6: **q**($\mathbf{n}$) – Real (Kind=nag_wp) array *Input*

On entry: q, the weights of the Gaussian sources.

7: **p1** – Integer *Input/Output*

On entry: p_1, the number of terms of the first Taylor series to be evaluated.

On exit: **p1** is unchanged.

Constraint: $\mathbf{p1} > 0$.

8: **p2** – Integer *Input/Output*

On entry: p_2, the number of terms of the second Taylor series to be evaluated.

On exit: **p2** is unchanged.

Constraint: $\mathbf{p2} > 0$.

9: **k** – Integer *Input/Output*

On entry: k, the number of clusters into which the source points will be aggregated.

On exit: **k** is unchanged.

Constraint: $1 \leq \mathbf{k} \leq \mathbf{n}$.

10: **hin**(**lhin**) – Real (Kind=nag_wp) array *Input*

On entry: h, the standard deviations of the Gaussian sources. If $\mathbf{lhin} < \mathbf{n}$, the array will be expanded automatically by repeating **hin** until it is of length **n**. See Section 2.6 in the G01 Chapter Introduction for further information.

Constraint: $\mathbf{hin}(i) > 0.0$, for $i = 1, 2, \ldots, \mathbf{lhin}$.

11: **lhin** – Integer *Input*

On entry: the length of the array **hin**.

Constraint: $1 \leq \mathbf{lhin} \leq \mathbf{n}$.

12: **tol** – Real (Kind=nag_wp) *Input*

On entry: ϵ, the desired accuracy of the FGT approximation of the DGT. Determines the radius of the source clusters: the contribution of a source point to the FGT approximation at a target point is disregarded if the source is outside the corresponding cluster radius.

Constraint: $\mathbf{tol} > 0.0$.

13: **v**(**m**) – Real (Kind=nag_wp) array *Output*

On exit: $\hat{G}(y)$, the value of the FGT evaluated at y.

14: **term**(**m**) – Real (Kind=nag_wp) array *Output*

On exit: **term**(j) contains the absolute value of the final Taylor series term that is largest, relative to the size of the sum of the corresponding series, across all clusters that contribute to the FGT at target point **v**(j).

15: **ifail** – Integer *Input/Output*

On entry: **ifail** must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of ifail on exit.**

On exit: **ifail** $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry **ifail** $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by **x04aaf**).

Errors or warnings detected by the routine:

ifail $= 1$

On entry, **d** $= \langle value \rangle$.
Constraint: **d** > 0.

ifail $= 2$

On entry, **n** $= \langle value \rangle$.
Constraint: **n** > 0.

ifail $= 3$

On entry, **m** $= \langle value \rangle$.
Constraint: **m** > 0.

ifail $= 4$

On entry, **p1** $= \langle value \rangle$.
Constraint: **p1** > 0.

ifail $= 5$

On entry, **p2** $= \langle value \rangle$.
Constraint: **p2** > 0.

ifail $= 6$

On entry, **k** $= \langle value \rangle$ and **n** $= \langle value \rangle$.
Constraint: $1 \leq$ **k** $\leq$ **n**.

ifail $= 7$

On entry, **hin**$(\langle value \rangle) = \langle value \rangle$.
Constraint: **hin**$(i) > 0.0$, for $i = 1, 2, \ldots,$ **lhin**.

ifail $= 8$

On entry, **lhin** $= \langle value \rangle$.
Constraint: $1 \leq$ **lhin** $\leq$ **n**.

ifail $= 9$

On entry, **tol** $= \langle value \rangle$.
Constraint: **tol** > 0.0.

ifail $= 10$

On exit, **p1** $= \langle value \rangle$, **p2** $= \langle value \rangle$ and **k** $= \langle value \rangle$.
p1, **p2** or **k** may have been too small to calculate $\mathbf{v}(\mathbf{m})$ to the required accuracy **tol**.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

ifail $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The routine does not currently implement the procedure described in Raykar and Duraiswami (2005) for automatically determining values for **p1**, **p2** and **k**. Non-zero values must therefore be provided for these parameters when calling the routine.

For a given set of source and target points and a specified tolerance, there is an interaction between the number of clusters, **k**, and the number of Taylor series terms, **p1** and **p2**: if the sources are clustered together in fewer clusters (small **k**) then more terms will be needed in each cluster's Taylor series (large **p1** and **p2**) to capture the effect of the source points further from the cluster centres. Increasing the number of clusters reduces their individual radii and requires fewer terms in their Taylor series, but increases the number of Taylor series that must be evaluated overall.

If the source and target points are uniformly distributed in a unit hypercube, Raykar and Duraiswami (2005) advise users to select $\mathbf{k} \sim \left\lceil (h_{\max} + h_{\min}/2)^{-d} \right\rceil$. If the points are not uniformly distributed then more clusters than this will be needed to calculate the FGT to within the specified **tol** without requiring prohibitively large values for **p1** and **p2**.

There is less guidance available for selecting good values for **p1** and **p2**. As the number of Taylor series terms is a major factor on the computation time taken by this routine, you are advised to initially try a small number, e.g. 20 or so, and then tune **p1** and **p2** up or down based on the values returned. Note that **p1** and **p2** are not required to have identical values.

To aid the selection of values for **p1**, **p2** and **k**, the routine returns in **term**(j) the absolute value of the final Taylor series term that is largest, relative to the size of the sum of the corresponding series, across all clusters that contribute to the FGT at target point j. If this value is larger than the requested **tol**, the routine will additionally return a non-zero **ifail** value and you are advised to re-run the routine with larger **p1**, **p2** or **k**.

8 Parallelism and Performance

c06saf is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

c06saf makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The time complexity of the algorithm implemented by this subroutine is $O(M+N)$, versus the $O(MN)$ time complexity of evaluating the DGT directly.

10 Example

In this example values for x, y, p_1, p_2, k and ϵ are read in, $\hat{G}(y)$ calculated and the results displayed.

10.1 Program Text

```
    Program c06safe

!     C06SAF Example Program Text

!     Mark 26.1 Release. NAG Copyright 2017.

!     .. Use Statements ..
      Use nag_library, Only: c06saf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                 :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)                 :: tol
      Integer                            :: d, ifail, ii, k, m, n, p1, p2
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable    :: hin(:), q(:), srcs(:,:), term(:),    &
                                            trgs(:,:), v(:)
!     .. Executable Statements ..
      Write (nout,*) 'C06SAF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) d, n, m

      Allocate (q(n))
      Allocate (hin(n))
      Allocate (srcs(d,n))
      Allocate (trgs(d,m))
      Allocate (v(m))
      Allocate (term(m))

      Read (nin,*) p1, p2, k
      Read (nin,*) tol
      Read (nin,*) q(1:n)
      Read (nin,*) hin(1:n)
      Read (nin,*) srcs(1:d,1:n)
      Read (nin,*) trgs(1:d,1:m)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call c06saf(d,srcs,n,trgs,m,q,p1,p2,k,hin,n,tol,v,term,ifail)
```

```
      Write (nout,*)
      Write (nout,*) '     y        FGT(y)      term'
      Write (nout,*)
      Write (nout,99999)(trgs(1:d,ii),v(ii),term(ii),ii=1,m)

99999 Format (2(1X,F4.1),3X,F8.3,4X,F8.6)

    End Program c06safe
```

10.2 Program Data

```
C06SAF Example Program Data
   2   5   5                                      : d, n, m
  10  10   1                                      : p1, p2, k
   0.001                                          : tol
   0.65 0.7 0.75 0.8 0.85                         : q
   0.9  1.0 1.1  1.2 1.3                          : hin
   0.0  0.0
   0.2  0.2
   0.4  0.4
   0.6  0.6
   0.8  0.8                                       : srcs
   0.1  0.0
   0.3  0.2
   0.5  0.4
   0.7  0.6
   0.9  0.8                                       : trgs
```

10.3 Program Results

```
 C06SAF Example Program Results

     y        FGT(y)      term

  0.1  0.0      2.877    0.000000
  0.3  0.2      3.231    0.000000
  0.5  0.4      3.256    0.000000
  0.7  0.6      2.985    0.000004
  0.9  0.8      2.518    0.000470
```