

# NAG Library Routine Document

## G01AGF

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

### 1 Purpose

G01AGF performs a scatter plot of two variables on a character printing device, with a chosen number of character positions in each direction.

### 2 Specification

```
SUBROUTINE G01AGF (X, Y, NOBS, ISORT, NSTEPX, NSTEPY, IFAIL)
INTEGER          NOBS, ISORT(NOBS), NSTEPX, NSTEPY, IFAIL
REAL (KIND=nag_wp) X(NOBS), Y(NOBS)
```

### 3 Description

G01AGF finds the range of the data in each dimension and calculates a step size for each division on the axes; these step sizes are selected from the list

$$(0.1, 0.15, 0.2, 0.25, 0.4, 0.5, 0.6, 0.75, 0.8) \times \text{power of } 10.$$

The axes are drawn and annotated and data points are plotted on the nearest character position. The character plotted is either a digit 1 to 9 for the equivalent number of occurrences of a point at a particular character position, an alphabetic A–Z for 10–35 occurrences, or \* if there are more than 35 coincident occurrences. Axes are drawn on all sides of the plot with the left-hand and bottom ones annotated; zero axes are also marked if included in the plotting area.

The Fortran logical unit number used for the output is the current advisory message unit number defined for each implementation. This number may be changed by an appropriate call to X04ABF before calling G01AGF.

### 4 References

None.

### 5 Arguments

- |    |  |                     |
|----|--|---------------------|
| 1: | X(NOBS) – REAL (KIND=nag_wp) array   | <i>Input</i>        |
|    | <i>On entry:</i> the values to be plotted in the <i>x</i> -direction.            |                     |
| 2: | Y(NOBS) – REAL (KIND=nag_wp) array   | <i>Input/Output</i> |
|    | <i>On entry:</i> the values to be plotted in the <i>y</i> -direction.            |                     |
|    | <i>On exit:</i> the elements of Y are sorted into descending order of magnitude. |                     |
| 3: | NOBS – INTEGER   | <i>Input</i>        |
|    | <i>On entry:</i> the number of observations to be plotted.                       |                     |
|    | <i>Constraint:</i> NOBS ≥ 1.   |                     |

- 4: ISORT(NOBS) – INTEGER array *Output*  
*On exit:* the key to the descending order of the elements in array Y, i.e., ISORT(*i*) contains the position of the value Y(*i*) in the original array Y, for  $i = 1, 2, \dots, \text{NOBS}$ .
- 5: NSTEPX – INTEGER *Input*  
*On entry:* the number of steps (character positions) to be plotted in the *x*-direction. If the supplied value of NSTEPX is less than 10, the value 10 will be used by G01AGF. The maximum value for NSTEPX is the number of character positions available on the chosen output device less 15, up to a maximum of 133. If NSTEPX exceeds 133 on input, the value 133 will be used by the routine.
- 6: NSTEPY – INTEGER *Input*  
*On entry:* the number of steps (character positions) to be plotted in the *y*-direction. If the supplied value of NSTEPY is less than 10, the value 10 will be used by G01AGF. There is no maximum value for NSTEPY, but you should bear in mind that (NSTEPY + 5) records (lines) of output are generated by the routine.
- 7: 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, NOBS < 1.

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

Accuracy is limited by the number of plotting points available.

## 8 Parallelism and Performance

G01AGF 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 routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

## 9 Further Comments

The time taken by G01AGF may be expected to be approximately proportional to the product  $\text{NOBS} \times \text{NSTEPX} \times \text{NSTEPI}$ .

No blank records are output before or after the plot.

You must make sure that it is permissible to write records containing NSTEPX characters to the current advisory message unit.

## 10 Example

The data relate to wheat and potato yields in 48 counties in England in 1936. The example illustrates the use of X04ABF to set the logical unit number, used for the output of G01AGF, to a specified value.

### 10.1 Program Text

```

Program g01agfe

!      G01AGF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: g01agf, nag_wp, x04abf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: iset = 1, nin = 5, nout = 6
!      .. Local Scalars ..
Integer                    :: i, ifail, nobs, nstepx, nstepy,      &
                          outchn
Character (80)             :: title
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: x(:), y(:)
Integer, Allocatable       :: isort(:)
!      .. Intrinsic Procedures ..
Intrinsic                  :: len_trim
!      .. Executable Statements ..
Write (nout,*) 'G01AGF Example Program Results'
Write (nout,*)

!      Skip heading in data file
Read (nin,*)

!      Read in the problem size
Read (nin,*) nobs, nstepx, nstepy

Allocate (x(nobs),y(nobs),isort(nobs))

!      Read in data
Read (nin,*)(x(i),y(i),i=1,nobs)

```

```

!      Read in the title
      Read (nin,99999) title

!      Set advisory channel
      outchn = nout
      Call x04abf(iset,outchn)

!      Display title
      Write (nout,*) title(1:len_trim(title))
      Write (nout,*)
      Flush (nout)

!      Produce the plot
      ifail = 0
      Call g01agf(x,y,nobs,isort,nstepx,nstepy,ifail)

99999 Format (A80)
      End Program g01agfe

```

## 10.2 Program Data

G01AGF Example Program Data

```

48 40 32
16.0 5.3 16.0 6.6 16.4 6.1 20.5 5.5 18.3 6.9 16.3 6.1
17.7 6.4 15.3 6.3 16.5 7.8 16.9 8.3 21.9 5.7 15.5 6.2
15.9 6.0 16.1 6.1 18.5 6.6 12.7 4.8 15.7 4.9 14.3 5.1
13.9 5.5 12.8 6.7 12.0 6.5 15.6 5.2 15.9 5.2 16.7 7.1
14.3 4.9 14.4 5.6 15.2 6.4 14.1 6.9 15.5 5.6 16.5 6.1
14.3 5.7 13.2 5.0 13.9 6.5 14.4 6.2 13.5 5.2 11.2 6.6
14.4 5.8 15.5 6.3 18.5 6.3 16.4 5.8 17.1 5.9 16.9 6.5
17.5 5.8 15.9 5.7 19.2 7.2 17.7 6.5 15.2 5.4 17.1 6.3
Plot of wheat (X) and potato (Y) yields in 48 counties in England in 1936.

```

### 10.3 Program Results

G01AGF Example Program Results

Plot of wheat (X) and potato (Y) yields in 48 counties in England in 1936.

