# NAG Fortran Library Routine Document

# G13BAF

**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

G13BAF filters a time series by an ARIMA model.

## 2    Specification

```
SUBROUTINE G13BAF(Y, NY, MR, NMR, PAR, NPAR, CY, WA, NWA, B, NB, IFAIL)
INTEGER          NY, MR(NMR), NMR, NPAR, NWA, NB, IFAIL
real             Y(NY), PAR(NPAR), CY, WA(NWA), B(NB)
```

## 3    Description

From a given series $y_1, y_2, \ldots, y_n$, a new series $b_1, b_2, \ldots, b_n$ is calculated using a supplied (filtering) ARIMA model. This model will be one which has previously been fitted to a series $x_t$ with residuals $a_t$. The equations defining $b_t$ in terms of $y_t$ are very similar to those by which $a_t$ is obtained from $x_t$. The only dissimilarity is that no constant correction is applied after differencing. This is because the series $y_t$ is generally distinct from the series $x_t$ with which the model is associated, though $y_t$ may be related to $x_t$. Whilst it is appropriate to apply the ARIMA model to $y_t$ so as to preserve the same relationship between $b_t$ and $a_t$ as exists between $y_t$ and $x_t$, the constant term in the ARIMA model is inappropriate for $y_t$. The consequence is that $b_t$ will not necessarily have zero mean.

The equations are precisely:

$$w_t = \nabla^d \nabla_s^D y_t, \tag{1}$$

the appropriate differencing of $y_t$; both the seasonal and non-seasonal inverted autoregressive operations are then applied,

$$u_t = w_t - \Phi_1 w_{t-s} - \cdots - \Phi_P w_{t-s \times P} \tag{2}$$

$$v_t = u_t - \phi_1 u_{t-1} - \cdots - \phi_p u_{t-p} \tag{3}$$

followed by the inverted moving average operations

$$z_t = v_t + \Theta_1 z_{t-s} + \cdots + \Theta_Q z_{t-s \times Q} \tag{4}$$

$$b_t = z_t + \theta_1 b_{t-1} + \cdots + \theta_q b_{t-q}. \tag{5}$$

Because the filtered series value $b_t$ depends on present and past values $y_t, y_{t-1}, \ldots$, there is a problem arising from ignorance of $y_0, y_{-1}, \ldots$ which particularly affects calculation of the early values $b_1, b_2, \ldots$, causing 'transient errors'. The routine allows two possibilities.

(i)   The equations (1), (2) and (3) are applied from successively later time points so that all terms on their right-hand sides are known, with $v_t$ being defined for $t = (1 + d + s \times D + s \times P), \ldots, n$. Equations (4) and (5) are then applied over the same range, taking any values on the right-hand side associated with previous time points to be zero.

This procedure may still however result in unacceptably large transient errors in early values of $b_t$.

(ii)  The unknown values $y_0, y_{-1}, \ldots$ are estimated by backforecasting. This requires that an ARIMA model distinct from that which has been supplied for filtering, should have been previously fitted to $y_t$.

For efficiency, the user is asked to supply both this ARIMA model for $y_t$ and a limited number of backforecasts which are prefixed to the known values of $y_t$. Within the routine further backforecasts of $y_t$, and the series $w_t$, $u_t$, $v_t$ in (1), (2), (3) are then easily calculated, and a set of linear equations solved for backforecasts of $z_t$, $b_t$ for use in (4) and (5) in the case that $q + Q > 0$.

Even if the best model for $y_t$ is not available, a very approximate guess such as

$$y_t = c + e_t$$

or

$$\nabla y_t = e_t$$

can help to reduce the transients substantially.

The backforecasts which need to be prefixed to $y_t$ are of length $Q'_y = q_y + s_y \times Q_y$, where $q_y$ and $Q_y$ are the non-seasonal and seasonal moving average orders and $s_y$ the seasonal period for the ARIMA model of $y_t$. Thus the user need not carry out the backforecasting exercise if $Q'_y = 0$. Otherwise, the series $y_1, y_2, \ldots, y_n$ should be reversed to obtain $y_n, y_{n-1}, \ldots, y_1$ and G13AJF should be used to forecast $Q'_y$ values, $\hat{y}_0, \ldots, \hat{y}_{1-Q'_y}$. The ARIMA model used is that fitted to $y_t$ (as a forward series) except that, if $d_y + D_y$ is odd, the constant should be changed in sign (to allow, for example, for the fact that a forward upward trend is a reversed downward trend). The ARIMA model for $y_t$ supplied to the filtering routine must however have the appropriate constant for the forward series.

The series $\hat{y}_{1-Q'_y}, \ldots, \hat{y}_0, y_1, \ldots, y_n$ is then supplied to the routine, and a corresponding set of values returned for $b_t$.

## 4    References

Box G E P and Jenkins G M (1976) *Time Series Analysis: Forecasting and Control* (Revised Edition) Holden-Day

## 5    Parameters

1:    Y(NY) – **real** array                                                                                          *Input*

*On entry*: the $Q'_y$ backforecasts, starting with backforecast at time $1 - Q'_y$ to backforecast at time 0, followed by the time series starting at time 1, where $Q'_y = \text{MR}(10) + \text{MR}(13) \times \text{MR}(14)$. If there are no backforecasts, either because the ARIMA model for the time series is not known, or because it is known but has no moving average terms, then the time series starts at the beginning of Y.

2:    NY – INTEGER                                                                                                   *Input*

*On entry*: the total number of backforecasts and time series data points in array Y.

*Constraint*: $\text{NY} \geq \max(1 + Q'_y, \text{NPAR})$.

3:    MR(NMR) – INTEGER array                                                                                     *Input*

*On entry*: the orders vector for the filtering model, followed by the orders vector for the ARIMA model for the time series if the latter is known. The orders appear in the standard sequence $(p, d, q, P, D, Q, s)$ as given in the G13 Chapter Introduction. If the ARIMA model for the time series is supplied, then the routine will assume that the first $Q'_y$ values of the array Y are backforecasts.

*Constraints*:   the filtering model is restricted in the following ways:

$\text{MR}(1) + \text{MR}(3) + \text{MR}(4) + \text{MR}(6) > 0$, i.e., filtering by a model which contains only differencing terms is not permitted;
$\text{MR}(k) \geq 0$, for $k = 1, 2, \ldots, 7$;
$\text{MR}(4) + \text{MR}(5) + \text{MR}(6) = 0$ if and only if $\text{MR}(7) = 0$;
$\text{MR}(7) \neq 1$.

the ARIMA model for the time series is restricted in the following ways:

$$\text{MR}(k) \geq 0, \text{ for } k = 8, \ldots, 14;$$
$$\text{MR}(11) + \text{MR}(12) + \text{MR}(13) = 0 \text{ if and only if } \text{MR}(14) = 0;$$
$$\text{MR}(14) \neq 1.$$

4:    NMR – INTEGER                                                                                                *Input*

*On entry*: the number of values specified in the array MR. It takes the value 7 if no ARIMA model for the time series is supplied but otherwise it takes the value 14. Thus NMR acts as an indicator as to whether backforecasting can be carried out.

*Constraint*: NMR = 7 or 14.

5:    PAR(NPAR) – ***real*** array                                                                                *Input*

*On entry*: the parameters of the filtering model, followed by the parameters of the ARIMA model for the time series, if supplied. Within each model the parameters are in the standard order of non-seasonal AR and MA followed by seasonal AR and MA.

6:    NPAR – INTEGER                                                                                              *Input*

*On entry*: the total number of parameters held in array PAR.

*Constraints*:

if NMR = 7, NPAR = MR(1) + MR(3) + MR(4) + MR(6),
if NMR = 14, NPAR = MR(1) + MR(3) + MR(4) + MR(6) + MR(8) + MR(10) +
MR(11) + MR(13).

**Note:** the first constraint (i.e., MR(1) + MR(3) + MR(4) + MR(6) > 0) on the orders of the filtering model, in parameter MR, ensures that NPAR > 0.

7:    CY – ***real***                                                                                             *Input*

*On entry*: if the ARIMA model is known (i.e., NMR = 14), CY must specify the constant term of the ARIMA model for the time series. If this model is not known (i.e., NMR = 7), then CY is not used.

8:    WA(NWA) – ***real*** array                                                                             *Workspace*
9:    NWA – INTEGER                                                                                              *Input*

*On entry*: the dimension of the array WA as declared in the (sub)program from which G13BAF is called. Workspace is only required if the ARIMA model for the time series is known.

*Constraints*:

let K = MR(3) + MR(6) × MR(7) + MR(8) + MR(9) + (MR(11) + MR(12)) × MR(14),
then
if NMR = 14, NWA ≥ K × (K + 2),
if NMR = 7, NWA ≥ 1.

10:    B(NB) – ***real*** array                                                                                 *Output*

*On exit*: the filtered output series. If the ARIMA model for the time series was known, and hence $Q_y'$ backforecasts were supplied in Y, then B contains $Q_y'$ 'filtered' backforecasts followed by the filtered series. Otherwise, the filtered series begins at the start of B just as the original series began at the start of Y. In either case, if the value of the series at time $t$ is held in Y($t$), then the filtered value at time $t$ is held in B($t$).

11:    NB – INTEGER                                                                                             *Input*

*On entry*: the dimension of the array B as declared in the (sub)program from which G13BAF is called. In addition to holding the returned filtered series, B is also used as an intermediate work array if the ARIMA model for the time series was known.

*Constraints*:

> let $K_1 = \text{MR}(1) + \text{MR}(4) \times \text{MR}(7)$, let $K_2 = \text{MR}(2) + \text{MR}(5) \times \text{MR}(7)$, and let
> $K_3 = \text{MR}(3) + \text{MR}(6) \times \text{MR}(7)$, then
> if $\text{NMR} = 14$, $\text{NB} \geq \text{NY} + \max(K_3, K_1 + K_2)$,
> if $\text{NMR} = 7$, $\text{NB} \geq \text{NY}$.

12:   IFAIL – INTEGER                                                         *Input/Output*

*On entry*: IFAIL must be set to 0, −1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

*On exit*: IFAIL = 0 unless the routine detects an error (see Section 6).

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, for users not familiar with this parameter the recommended value is 0. **When the value −1 or 1 is used it is essential to test the value of IFAIL on exit.**

# 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, $\text{NMR} \neq 7$ and $\text{NMR} \neq 14$.

IFAIL = 2

> On entry, the orders vector MR does not satisfy the constraints given in Section 5.

IFAIL = 3

> On entry, NPAR is inconsistent with the contents of MR (see Section 5).

IFAIL = 4

> On entry, NY is too small to successfully carry out the requested filtering, (see Section 5).

IFAIL = 5

> On entry, the work array WA is too small.

IFAIL = 6

> On entry, the array B is too small.

IFAIL = 7

> The orders vector for the filtering model is invalid.

IFAIL = 8

> The orders vector for the ARIMA model is invalid. (Only occurs if $\text{NMR} = 14$.)

IFAIL = 9

> The initial values of the filtered series are indeterminate for the given models.

# 7   Accuracy

Accuracy and stability are high except when the MA parameters are close to the invertibility boundary.

## 8      Further Comments

The time taken by the routine is approximately proportional to

$$NY \times (MR(1) + MR(3) + MR(4) + MR(6)),$$

with an appreciable fixed increase if an ARIMA model is supplied for the time series.

## 9      Example

The example program reads a time series of length 296. It reads the univariate ARIMA (4,0,2,0,0,0,0) model and the ARIMA filtering (3,0,0,0,0,0,0) model for the series. Two initial backforecasts are required and these are calculated by a call to G13AJF. The backforecasts are inserted at the start of the series and G13BAF is called to perform the calculations.

### 9.1    Program Text

**Note:** the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*       G13BAF Example Program Text
*       Mark 14 Revised.  NAG Copyright 1989.
*       .. Parameters ..
        INTEGER          NXMAX, NPMAX, ISTMAX, IFVMAX, IW, IQXDMX, NYMAX,
       +                 NBMAX
        PARAMETER        (NXMAX=300,NPMAX=10,ISTMAX=20,IFVMAX=2,IW=2200,
       +                 IQXDMX=5,NYMAX=NXMAX+IQXDMX,NBMAX=320)
        INTEGER          NIN, NOUT
        PARAMETER        (NIN=5,NOUT=6)
*       .. Local Scalars ..
        real             A1, A2, CX, CY, RMS
        INTEGER          I, IDD, IDW, IFAIL, II, IJ, IQXD, J, K, N, NB,
       +                 NI, NMR, NPAR, NPARX, NST, NWA, NX, NY
*       .. Local Arrays ..
        real             B(NBMAX), FSD(IFVMAX), FVA(IFVMAX), PAR(NPMAX),
       +                 PARX(NPMAX), ST(ISTMAX), WA(IW), X(NXMAX),
       +                 Y(NYMAX)
        INTEGER          ISF(4), MR(14), MRX(7)
*       .. External Subroutines ..
        EXTERNAL         G13AJF, G13BAF
*       .. Intrinsic Functions ..
        INTRINSIC        MAX, MIN, MOD
*       .. Executable Statements ..
        WRITE (NOUT,*) 'G13BAF Example Program Results'
*       Skip heading in data file
        READ (NIN,*)
        READ (NIN,*) NX
        IF (NX.GT.0 .AND. NX.LE.NXMAX) THEN
           READ (NIN,*) (X(I),I=1,NX)
*          Read univariate ARIMA for series
           READ (NIN,*) (MRX(I),I=1,7)
           READ (NIN,*) CX
           NPARX = MRX(1) + MRX(3) + MRX(4) + MRX(6)
           IF (NPARX.GT.0 .AND. NPARX.LE.NPMAX) THEN
              READ (NIN,*) (PARX(I),I=1,NPARX)
*             Read model by which to filter series
              READ (NIN,*) (MR(I),I=1,7)
              NPAR = MR(1) + MR(3) + MR(4) + MR(6)
              IF (NPAR.GT.0 .AND. NPAR+NPARX.LE.NPMAX) THEN
                 READ (NIN,*) (PAR(I),I=1,NPAR)
*                Initially backforecast QY values
*                (1) Reverse series in situ
                 N = NX/2
                 NI = NX
                 DO 20 I = 1, N
                    A1 = X(I)
                    A2 = X(NI)
                    X(I) = A2
```

```
                         X(NI) = A1
                         NI = NI - 1
   20             CONTINUE
                  IDD = MRX(2) + MRX(5)
*                 (2) Possible sign reversal for ARIMA constant
                  IF (MOD(IDD,2).NE.0) CX = -CX
*                 (3) Calculate number of backforecasts required
                  IQXD = MRX(3) + MRX(6)*MRX(7)
                  IF (IQXD.NE.0) THEN
*                     (4) Set up parameter list for call to forecast
*                     routine G13AJF
                      IDW = IW
                      IFAIL = 0
*
                      CALL G13AJF(MRX,PARX,NPARX,CX,1,X,NX,RMS,ST,ISTMAX,
     +                           NST,IQXD,FVA,FSD,IFVMAX,ISF,WA,IW,IFAIL)
*
                      J = IQXD
                      DO 40 I = 1, IQXD
                         Y(I) = FVA(J)
                         J = J - 1
   40                 CONTINUE
*                     Move series into Y
                      J = IQXD + 1
                      K = NX
                      DO 60 I = 1, NX
                         IF (J.GT.NYMAX) STOP
                         Y(J) = X(K)
                         J = J + 1
                         K = K - 1
   60                 CONTINUE
                  END IF
*                 Calculate series length
                  NY = NX + IQXD
*                 Move ARIMA for series into MR
                  DO 80 I = 1, 7
                     MR(7+I) = MRX(I)
   80             CONTINUE
*                 Move parameters of ARIMA for Y into PAR
                  DO 100 I = 1, NPARX
                     PAR(NPAR+I) = PARX(I)
  100             CONTINUE
                  NPAR = NPAR + NPARX
*                 Move constant and reset sign reversal
                  CY = CX
                  IF (MOD(IDD,2).NE.0) CY = -CY
*                 Set parameters for call to filter routine G13BAF
                  NMR = 14
                  NWA = MR(3) + MR(6)*MR(7) + MR(8) + MR(9) + (MR(11)
     +                  +MR(12))*MR(14)
                  NWA = NWA*(NWA+2)
                  NB = NY + MAX(MR(3)+MR(6)*MR(7),MR(1)+MR(2)+(MR(4)+MR(5))
     +                  *MR(7))
                  IF (NWA.LE.IW .AND. NB.LE.NBMAX) THEN
                      IFAIL = 0
*                     Filter series by call to G13BAF
                      CALL G13BAF(Y,NY,MR,NMR,PAR,NPAR,CY,WA,NWA,B,NB,IFAIL)
*
                      WRITE (NOUT,*)
                      WRITE (NOUT,*)
     +                  '                  Original       Filtered'
                      WRITE (NOUT,*)
     +                  'Backforecasts    y-series        series'
                      IF (IQXD.NE.0) THEN
                          IJ = -IQXD
                          DO 120 I = 1, IQXD
                             WRITE (NOUT,99999) IJ, Y(I), B(I)
                             IJ = IJ + 1
  120                     CONTINUE
                      END IF
                      WRITE (NOUT,*)
```

```
                  WRITE (NOUT,*)
      + '         Filtered        Filtered        Filtered        Filtered'
                  WRITE (NOUT,*)
      + '          series          series          series          series'
                  DO 140 I = IQXD + 1, NY, 4
                     WRITE (NOUT,99998) (II-IQXD,B(II),II=I,MIN(NY,I+3))
  140             CONTINUE
               END IF
            END IF
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,I8,F17.4,F15.4)
99998 FORMAT (1X,I5,F9.4,I7,F9.4,I7,F9.4,I7,F9.4)
      END
```

## 9.2   Program Data

```
G13BAF Example Program Data
   296
 53.8 53.6 53.5 53.5 53.4 53.1 52.7 52.4 52.2 52.0 52.0 52.4 53.0 54.0 54.9 56.0
 56.8 56.8 56.4 55.7 55.0 54.3 53.2 52.3 51.6 51.2 50.8 50.5 50.0 49.2 48.4 47.9
 47.6 47.5 47.5 47.6 48.1 49.0 50.0 51.1 51.8 51.9 51.7 51.2 50.0 48.3 47.0 45.8
 45.6 46.0 46.9 47.8 48.2 48.3 47.9 47.2 47.2 48.1 49.4 50.6 51.5 51.6 51.2 50.5
 50.1 49.8 49.6 49.4 49.3 49.2 49.3 49.7 50.3 51.3 52.8 54.4 56.0 56.9 57.5 57.3
 56.6 56.0 55.4 55.4 56.4 57.2 58.0 58.4 58.4 58.1 57.7 57.0 56.0 54.7 53.2 52.1
 51.6 51.0 50.5 50.4 51.0 51.8 52.4 53.0 53.4 53.6 53.7 53.8 53.8 53.8 53.3 53.0
 52.9 53.4 54.6 56.4 58.0 59.4 60.2 60.0 59.4 58.4 57.6 56.9 56.4 56.0 55.7 55.3
 55.0 54.4 53.7 52.8 51.6 50.6 49.4 48.8 48.5 48.7 49.2 49.8 50.4 50.7 50.9 50.7
 50.5 50.4 50.2 50.4 51.2 52.3 53.2 53.9 54.1 54.0 53.6 53.2 53.0 52.8 52.3 51.9
 51.6 51.6 51.4 51.2 50.7 50.0 49.4 49.3 49.7 50.6 51.8 53.0 54.0 55.3 55.9 55.9
 54.6 53.5 52.4 52.1 52.3 53.0 53.8 54.6 55.4 55.9 55.9 55.2 54.4 53.7 53.6 53.6
 53.2 52.5 52.0 51.4 51.0 50.9 52.4 53.5 55.6 58.0 59.5 60.0 60.4 60.5 60.2 59.7
 59.0 57.6 56.4 55.2 54.5 54.1 54.1 54.4 55.5 56.2 57.0 57.3 57.4 57.0 56.4 55.9
 55.5 55.3 55.2 55.4 56.0 56.5 57.1 57.3 56.8 55.6 55.0 54.1 54.3 55.3 56.4 57.2
 57.8 58.3 58.6 58.8 58.8 58.6 58.0 57.4 57.0 56.4 56.3 56.4 56.4 56.0 55.2 54.0
 53.0 52.0 51.6 51.6 51.1 50.4 50.0 50.0 52.0 54.0 55.1 54.5 52.8 51.4 50.8 51.2
 52.0 52.8 53.8 54.5 54.9 54.9 54.8 54.4 53.7 53.3 52.8 52.6 52.6 53.0 54.3 56.0
 57.0 58.0 58.6 58.5 58.3 57.8 57.3 57.0
     4     0     2     0     0     0     0
   0.000
   2.420    -2.380     1.160    -0.230     0.310    -0.470
     3     0     0     0     0     0     0
   1.970    -1.370     0.340
```

## 9.3   Program Results

```
G13BAF Example Program Results


                  Original        Filtered
Backforecasts     y-series        series
      -2          49.9807         3.4222
      -1          52.6714         3.0809


      Filtered          Filtered          Filtered          Filtered
      series            series            series            series
   1  2.9813       2  2.7803       3  3.7057       4  3.2450
   5  3.0760       6  3.0070       7  3.0610       8  3.1720
   9  3.1170      10  3.0360      11  3.2580      12  3.4520
  13  3.3320      14  3.6980      15  3.3140      16  3.8070
  17  3.3330      18  2.9580      19  3.2800      20  3.0960
  21  3.2270      22  3.0830      23  2.6410      24  3.1870
  25  2.9910      26  3.1110      27  2.8460      28  3.0240
  29  2.7030      30  2.6130      31  2.8060      32  2.9560
  33  2.8170      34  2.8950      35  2.8510      36  2.9160
  37  3.2530      38  3.3050      39  3.1830      40  3.3760
  41  2.9730      42  2.8610      43  3.0490      44  2.8420
  45  2.3190      46  2.3660      47  2.9410      48  2.3810
  49  3.3420      50  2.9340      51  3.1800      52  2.9230
```

| 53 | 2.6470 | 54 | 2.8860 | 55 | 2.5310 | 56 | 2.6200 |
| 57 | 3.4170 | 58 | 3.4940 | 59 | 3.2590 | 60 | 3.1310 |
| 61 | 3.1420 | 62 | 2.6710 | 63 | 2.8990 | 64 | 2.8180 |
| 65 | 3.2150 | 66 | 2.8800 | 67 | 2.9610 | 68 | 2.8800 |
| 69 | 3.0020 | 70 | 2.8930 | 71 | 3.1210 | 72 | 3.2210 |
| 73 | 3.2040 | 74 | 3.5360 | 75 | 3.7520 | 76 | 3.5630 |
| 77 | 3.7260 | 78 | 3.1560 | 79 | 3.6310 | 80 | 2.9380 |
| 81 | 3.1480 | 82 | 3.4490 | 83 | 3.1400 | 84 | 3.7380 |
| 85 | 4.1200 | 86 | 3.1540 | 87 | 3.7480 | 88 | 3.3280 |
| 89 | 3.3640 | 90 | 3.3400 | 91 | 3.3950 | 92 | 3.0720 |
| 93 | 3.0050 | 94 | 2.8520 | 95 | 2.7810 | 96 | 3.1950 |
| 97 | 3.2490 | 98 | 2.6370 | 99 | 3.0080 | 100 | 3.2410 |
| 101 | 3.5570 | 102 | 3.2080 | 103 | 3.0880 | 104 | 3.3980 |
| 105 | 3.1660 | 106 | 3.1960 | 107 | 3.2460 | 108 | 3.2870 |
| 109 | 3.1590 | 110 | 3.2620 | 111 | 2.7280 | 112 | 3.4130 |
| 113 | 3.2190 | 114 | 3.6750 | 115 | 3.8550 | 116 | 4.0100 |
| 117 | 3.5380 | 118 | 3.8440 | 119 | 3.4660 | 120 | 3.0640 |
| 121 | 3.4780 | 122 | 3.1140 | 123 | 3.5300 | 124 | 3.2400 |
| 125 | 3.3630 | 126 | 3.2610 | 127 | 3.3020 | 128 | 3.1150 |
| 129 | 3.3280 | 130 | 2.8730 | 131 | 3.0800 | 132 | 2.8390 |
| 133 | 2.6570 | 134 | 3.0260 | 135 | 2.4580 | 136 | 3.2600 |
| 137 | 2.8380 | 138 | 3.2150 | 139 | 3.1140 | 140 | 3.1050 |
| 141 | 3.1400 | 142 | 2.9100 | 143 | 3.1370 | 144 | 2.7500 |
| 145 | 3.1160 | 146 | 3.0680 | 147 | 2.8590 | 148 | 3.3840 |
| 149 | 3.5500 | 150 | 3.4160 | 151 | 3.1770 | 152 | 3.3390 |
| 153 | 3.0190 | 154 | 3.1780 | 155 | 3.0110 | 156 | 3.1940 |
| 157 | 3.2680 | 158 | 3.0500 | 159 | 2.8060 | 160 | 3.1850 |
| 161 | 3.0560 | 162 | 3.2690 | 163 | 2.7940 | 164 | 3.0900 |
| 165 | 2.7100 | 166 | 2.7890 | 167 | 2.9510 | 168 | 3.2440 |
| 169 | 3.2570 | 170 | 3.4360 | 171 | 3.4450 | 172 | 3.3780 |
| 173 | 3.3520 | 174 | 3.9180 | 175 | 2.9190 | 176 | 3.1780 |
| 177 | 2.2580 | 178 | 3.5150 | 179 | 2.8010 | 180 | 3.6030 |
| 181 | 3.2610 | 182 | 3.5300 | 183 | 3.3270 | 184 | 3.4420 |
| 185 | 3.5240 | 186 | 3.2720 | 187 | 3.1110 | 188 | 2.8240 |
| 189 | 3.2330 | 190 | 3.1500 | 191 | 3.5710 | 192 | 3.0810 |
| 193 | 2.7820 | 194 | 2.9040 | 195 | 3.2350 | 196 | 2.7970 |
| 197 | 3.1320 | 198 | 3.1680 | 199 | 4.5210 | 200 | 2.6650 |
| 201 | 4.6870 | 202 | 3.9470 | 203 | 3.2220 | 204 | 3.3410 |
| 205 | 3.9950 | 206 | 3.4820 | 207 | 3.3630 | 208 | 3.4550 |
| 209 | 3.2950 | 210 | 2.6910 | 211 | 3.4600 | 212 | 2.9440 |
| 213 | 3.4400 | 214 | 3.1830 | 215 | 3.4200 | 216 | 3.4100 |
| 217 | 4.0550 | 218 | 2.9990 | 219 | 3.8250 | 220 | 3.1340 |
| 221 | 3.5010 | 222 | 3.0430 | 223 | 3.2660 | 224 | 3.3660 |
| 225 | 3.2650 | 226 | 3.3720 | 227 | 3.2880 | 228 | 3.5470 |
| 229 | 3.6840 | 230 | 3.3100 | 231 | 3.6790 | 232 | 3.1780 |
| 233 | 2.9360 | 234 | 2.7910 | 235 | 3.8020 | 236 | 2.6100 |
| 237 | 4.1690 | 238 | 3.7460 | 239 | 3.4560 | 240 | 3.3910 |
| 241 | 3.5820 | 242 | 3.6220 | 243 | 3.4870 | 244 | 3.5770 |
| 245 | 3.4240 | 246 | 3.3960 | 247 | 3.1220 | 248 | 3.4300 |
| 249 | 3.4580 | 250 | 3.0280 | 251 | 3.7660 | 252 | 3.3770 |
| 253 | 3.2470 | 254 | 3.0180 | 255 | 2.9720 | 256 | 2.8000 |
| 257 | 3.2040 | 258 | 2.8020 | 259 | 3.4100 | 260 | 3.1680 |
| 261 | 2.4600 | 262 | 2.8810 | 263 | 3.1750 | 264 | 3.1740 |
| 265 | 4.8640 | 266 | 3.0600 | 267 | 2.9600 | 268 | 2.2530 |
| 269 | 2.5620 | 270 | 3.3150 | 271 | 3.3480 | 272 | 3.5900 |
| 273 | 3.2560 | 274 | 3.2320 | 275 | 3.6160 | 276 | 3.1700 |
| 277 | 3.2890 | 278 | 3.1200 | 279 | 3.3300 | 280 | 2.9910 |
| 281 | 2.9420 | 282 | 3.4070 | 283 | 2.8720 | 284 | 3.3470 |
| 285 | 3.1920 | 286 | 3.4880 | 287 | 4.0680 | 288 | 3.7550 |
| 289 | 3.0510 | 290 | 3.9680 | 291 | 3.3900 | 292 | 3.1380 |
| 293 | 3.6170 | 294 | 3.1700 | 295 | 3.4150 | 296 | 3.4830 |