

Subject: Feature creep in Clause 13
 From: Van Snyder

1 Edits

2 Edits refer to 07-007. Page and line numbers are displayed in the margin. Absent other instructions, a
 3 page and line number or line number range implies all of the indicated text is to be replaced by associated
 4 text, while a page and line number followed by + (-) indicates that associated text is to be inserted after
 5 (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

6 1.1 More Bessel functions

7 Bessel functions of general order are usually computed using a Miller algorithm. Therefore, to compute
 8 $J_n(x)$ it is necessary to compute $J_{n-1}(x)$. Furthermore, Bessel functions of several consecutive orders
 9 are frequently needed, for example for Neumann expansions.

10	[Editor: Add “or BESSEL_JN (N1,N2,X)”.]	359:8
11	Class.	359:10
12	<i>Case (i):</i> BESSEL_JN (N,X) is elemental.	
13	<i>Case (ii):</i> BESSEL_JN (N1,N2,X) is transformational.	
14	N1 shall be of type integer and nonnegative.	359:12+
15	N2 shall be of type integer and nonnegative.	
16	Result Characteristics. Same type and kind as X.	359:14
17	<i>Case (i):</i> The result of BESSEL_JN (N,X) is scalar.	
18	<i>Case (ii):</i> The result of BESSEL_JN (N1,N2,X) is a rank-one array with extent MAX(N2–N1+1,0).	
19	Result Value.	359:15-16
20	<i>Case (i):</i> The result value of BESSEL_JN (N,X) is a processor-dependent approximation	
21	to the Bessel function of the first kind of order N of X.	
22	<i>Case (ii):</i> Element <i>i</i> of the result value of BESSEL_JN (N1,N2,X) is a processor-dependent	
23	approximation to the Bessel function of the first kind of order N1+ <i>i</i> – 1 of X.	
24	[Editor: Add “or BESSEL_YN (N1,N2,X)”.]	360:2
25	Class.	360:4
26	<i>Case (i):</i> BESSEL_YN (N,X) is elemental.	
27	<i>Case (ii):</i> BESSEL_YN (N1,N2,X) is transformational.	
28	N1 shall be of type integer and nonnegative.	360:6+
29	N2 shall be of type integer and nonnegative.	
30	Result Characteristics. Same type and kind as X.	360:8
31	<i>Case (i):</i> The result of BESSEL_YN (N,X) is scalar.	
32	<i>Case (ii):</i> The result of BESSEL_YN (N1,N2,X) is a rank-one array with extent MAX(N2–N1+1,0).	
33	Result Value.	360:9-10
34	<i>Case (i):</i> The result value of BESSEL_YN (N,X) is a processor-dependent approximation	
35	to the Bessel function of the second kind of order N of X.	
36	<i>Case (ii):</i> Element <i>i</i> of the result value of BESSEL_YN (N1,N2,X) is a processor-dependent	
37	approximation to the Bessel function of the second kind of order N1+ <i>i</i> – 1 of X.	

38 **1.2 Embellishment of NORM2**

39 [Editor: Insert “[, DIM]” after “X”.] 411:16

40 **Arguments.** 411:19-23

41 X shall be a real array.

42 DIM (optional) shall be an integer scalar. The corresponding actual argument shall not be an optional dummy argument

43 **Result Characteristics.** The result is of the same type and type parameters as X. It is scalar if DIM is absent; otherwise the result has rank $n - 1$ and shape $[d_1, d_2, \dots, d_{\text{DIM}-1}, d_{\text{DIM}+1}, \dots, d_n]$, where n is the rank of X and $[d_1, d_2, \dots, d_n]$ is the shape of X.]46 **Result Value.**47 *Case (i):* The result of NORM2(X) has a value equal to a processor-dependent approximation to the generalized L_2 norm of X, which is the square root of the sum of the squares of the elements of X.48 *Case (ii):* The result of NORM2(X,DIM=DIM) has a value equal to that of NORM2(X) if X has rank one. Otherwise, the value of element $(s_1, s_2, \dots, s_{\text{DIM}-1}, s_{\text{DIM}+1}, \dots, s_n)$ of the result is equal to NORM2($X(s_1, s_2, \dots, s_{\text{DIM}-1}, :, s_{\text{DIM}+1}, \dots, s_n)$).

53 It is recommended that the processor compute the result without undue overflow or underflow.

54 [Editor: Append “If X has the value $\begin{bmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \end{bmatrix}$ then NORM(X,DIM=1) is [3.162, 4.472] (approximately) and NORM(X,DIM=2) is [2.236, 5.0] (approximately).”] 411:2456 **2 Not part of passed edits**57 **2.1 Another shift function**

58 We have SHIFT, SHIFTL and SHIFTR, but only DSHIFTL and DSHIFTR.

59 **13.7.55 $\frac{1}{2}$ DSHIFT (I,J,SHIFT)** 375:26+60 **Description.** Combined shift.61 **Class.** Elemental function.62 **Arguments.**

63 I shall be of type integer or bits.

64 J shall be of type integer or bits.

65 SHIFT shall be of type integer. Its magnitude shall be less than or equal to BIT_SIZE(I).

66 **Result Characteristics.** Same as I.67 **Result Value.** If SHIFT is nonnegative the result value is the same as DSHIFTL(I,J,SHIFT).
68 If SHIFT is negative the result value is the same as DSHIFTR(I,J,-SHIFT).