

Subject: Updating real and imaginary parts of complex variables
 From: Van Snyder
 Reference: 03-258r1, section 2.2.5

1 **Number**

2 TBD

3 **Title**

4 Updating real and imaginary parts of complex variables.

5 **Submitted By**

6 J3

7 **Status**

8 For consideration.

9 **Basic Functionality**

10 Provide a way to update real and imaginary parts of complex variables independently of updating the
 11 whole variable.

12 **Rationale**

13 In some applications it is necessary to update only the real or imaginary part of a complex variable. To
 14 change the real part of a complex variable `C`, one currently needs to write

```
15 C = cmplx(new_real,aimag(C))
```

16 This doesn't look too bad, but consider the case of a complicated reference:

```
17 MyThing(I,23*(J-11*K),Func(M,I,J,K))%MyField(L+3)%Radiance(1:2,1:2) = &  
18   & cmplx(new_real,&  
19   & aimag(MyThing(I,23*(J-11*K),Func(M,I,J,K))%MyField(L+3)%Radiance(1:2,1:2)))
```

20 A processor might deduce that the LHS and the argument of `aimag` are the same (but it might be
 21 confused by the apparent need to invoke `Func` twice if `Func` isn't pure), but you must admit that the
 22 cost of maintenance is higher and its reliability lower than with the simpler alternative proposed below.
 23 One could simplify this a little bit by writing

```
24 ASSOCIATE ( &  
25   & C => MyThing(I,23*(J-11*K),Func(M,I,J,K))%MyField(L+3)%Radiance(1:2,1:2) )  
26   C = cmplx(new_real,aimag(C))  
27 END ASSOCIATE
```

28 but this is still ickier than necessary.

29 **Estimated Impact**

30 Minor.

1 Detailed Specification

2 It would be nicer if COMPLEX were a parameterized sequence derived type having two components, the
3 real part and the imaginary part, in that order called, say, REAL and IMAG. Then the above assignment
4 would be simplified to

```
5 C%real = new_real
```

6 or

```
7 MyThing(I,23*(J-11*K),Func(M,I,J,K))%MyField(L+3)%Radiance(1:2,1:2)%real = &  
8 & new_real
```

9 A side effect of this change is that there would be a new constructor for objects of complex type, named
10 COMPLEX. Its syntax of usage would be the same as for other derived-type constructors. This is a
11 desirable side effect. It is probably desirable to specify that if either component is not specified it has
12 a default value of zero; doing so would make the COMPLEX constructor work more like the Cmplx
13 intrinsic function.

14 This proposal would interact with the proposal to regularize type reference, i.e., if COMPLEX is defined
15 to be a sequence derived type, it would be reasonable to denote it by TYPE(COMPLEX).

16 If the proposal for accessor procedures is adopted this proposal will be unnecessary. One feature of that
17 proposal is that the AIMAG, Cmplx and REAL intrinsic functions ought to be an accessors. I.e.,
18 updating the real part of a complex number would be written

```
19 real(c) = new_real
```

20 or

```
21 real(MyThing(I,23*(J-11*K),Func(M,I,J,K))%MyField(L+3)%Radiance(1:2,1:2)) = &  
22 & new_real
```

23 History