Subject: INTERPOLATE note
Posted by Robert S. Hill on Thu, 18 Oct 2001 00:07:45 GMT

View Forum Message <> Reply to Message

This may have been remarked on before, but the built-in INTERPOLATE function seems to treat its "floating-point subscript" as single precision no matter what.  I don't see anything about this in the documentation.  For example,

```
p = [1.0d0,2.0d0]
x = 0.123123123123123d0
print,interpolate(p,x),format='(E25.15)'
   1.123123124241829E+00
print,interpolate(p,float(x)),format='(E25.15)'
   1.123123124241829E+00
```

Whereas, using the library routine INTERPOL,

```
print,interpol(p,[0.d0,1.d0],x),format='(E25.15)'
   1.123123123123123E+00
print,interpol(p,[0.d0,1.d0],float(x)),format='(E25.15)'
   1.123123124241829E+00
```

The first INTERPOL result is what I expected from both routines.  The second INTERPOL result is the same as what INTERPOLATE gives, and it shows that INTERPOLATE converts the "subscript" argument to single-precision.

I personally think it would be more straightforward if built-in routines did their default typecasting in the same way as ordinary arithmetic expressions, and left up to me the question of whether I really need all those digits of precision.  (In my application, as a matter of fact I probably don't.)

- Bob

--
```
   --------------------------------------------
     Robert S. Hill       301-286-3577
     Robert.S.Hill.1@gsfc.nasa.gov
     Science Systems and Applications, Inc.
     Code 681, NASA GSFC, Greenbelt, MD 20771
   --------------------------------------------
```