Subject: how to pass a complex variable to fortran subroutine
Posted by mvukovic on Mon, 08 Dec 2003 20:38:59 GMT

View Forum Message <> Reply to Message

Hoping to save myself some time, I ask the following question: how to pass a complex variable to fortran subroutine. Here is some background:

I am linking IDL to a fortran routine for the calculation of a Bessel function for a complex argument. This is on Windows 2000, using Cygwin and the g77/gcc compiler suite.

The routine input and output are complex. I am currently dealing with this with two fortran ``glue'' routines. Call_External passes the the real and imaginary components to the first glue routine which passes them directly to a second one. This second one creates a complex variable, and calls the fortran routine. (I am including the two routines at the end of the post).

The result from the Fortran procedure is also complex. In the second routine, it is decomposed into it's real and imaginary parts, which are then returned to the first glue routine, and back to IDL as components of a complex number.

I was unable to figure out how to compress this process into a single fortran routine. More specifically, how to pass the resulting components up to IDL.

(Something tells me that C should be able to do it in sleep, but C is not my forte.)

Another approach would be to pass a complex variable from IDL directly to Fortran, but I have crashed IDL about 50 times so far, and since I finally got it to work, I am a bit tired of experimenting. So, has anyone succesfully passed an IDL complex variable directly to a Fortran complex variable?

Thanks for thoughts and inputs.

Mirko

And now for the two glue routines:

```
C*****First glue routine, called directly from IDL
C     Interface routine between the IDL CBESI.PRO to CBESI.F  The
C     translation occurs in two stages. In this routine, all IDL
```

```
C     variables are passed through.  CBESI_WO_CMPLX will deal with
C     conversion of variables to from their real and imaginary
C     components to the COMPLEX form
      SUBROUTINE CBESI_DRIVER(argc, argv)
      INTEGER*4 argc, argv(*)

      CALL CBESI_WO_CMPLX(%VAL(argv(1)),%VAL(argv(2)), %VAL(argv(3)),
     1 %VAL(argv(4)),%VAL(argv(5)),%VAL(argv(6)), %VAL(argv(7)),
     2 %VAL(argv(8)))

      RETURN
      END

C**** Second glue routine.
C     This routine accepts and returns the arguments of CBESI, with
the
C     slight wrinkle that all complex arguments are decomposed into
C     their real and imaginary parts.  The reason for that is that I
am
C     uncertain how to pass complex idl variables to complex fortran
C     variables.
C
C     Furthermore, this routine does not allow for calculations for
C     multiple FNU's.  It explicitly calls CBESI with N = 1

      SUBROUTINE CBESI_WO_CMPLX(REZ, IMZ, FNU, KODE, RECY, IMCY,
     * NZ, IERR)

      COMPLEX*8 CY, Z
      REAL*4 FNU, REZ, IMZ, RECY, IMCY
      INTEGER*2 IERR, KODE, NZ

      Z=COMPLEX(REZ,IMZ)
      CALL CBESI(Z,FNU,KODE,1,CY,NZ,IERR)
C     REALPART and IMAGPART seem to be GNUFortran conventions.
      RECY=REALPART(CY)
      IMCY=IMAGPART(CY)


      RETURN
      END
```