Subject: Re: Linking IDL with FORTRAN routines under UNIX: interpol.pro
Posted by thompson on Tue, 24 Sep 1991 20:03:03 GMT
View Forum Message <> Reply to Message

This is the IDL procedure for the sample FORTRAN program for a Sun workstation.
The environment variable SERTS_EXTERNAL points to where the executable file
resides, and is just a convenience for myself.

```
 FUNCTION INTERPOL, V, X, U
;+
; NAME:
; INTERPOL
; PURPOSE:
; Linear interpolation for vectors.
; Regular or irregular grid.
; CATEGORY:
; E1 - Interpolation
; CALLING SEQUENCE:
; Result = INTERPOL( V, X, U) ;Irregular grids
; Result = INTERPOL( V, N) ;Regular grids
; INPUTS:
; V = input vector, should be one dimensional,
;  any type except string.
;
; Regular grids:
; N = Number of points of result, both input and
;  output grids are regular.  Output grid
;  absicissa value = float(i)/N_elements(V),
;  for i=0,n-1.
;
; Irregular grids:
; X = Absicissae values for V. Must have same # of
;  elements as V. MUST be monotonic, either
;  ascending or descending.
; U = Absicissae values for result.  Result will
;  have same number of elements as U.  U need
;  not be monotonic.
;
; OPTIONAL INPUT PARAMETERS:
; None.
; OUTPUTS:
; Result = Floating vector of N points determined
;  from linearly interpolating input vector.
;  If V is double or complex, result is double
;  or complex.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
```

```
; None.
; RESTRICTIONS:
; None.
; PROCEDURE:
; Result(i) = V(x) + (x-FIX(x))*(V(x+1)-V(x))
;  where x = i*(m-1)/(N-1) for regular grids.
;  m = # of elements in V, i=0 to N-1.
;  For irregular grids, x = U(i).
;  m = number of points of input vector.
;
; MODIFICATION HISTORY:
; Written, DMS, October, 1982.
; Modified, Rob at NCAR, February, 1991.  Made larger arrays possible
;  and correct by using long indexes into the array instead of
;  integers.
; Modified, William Thompson, August 1991, do interpolation via Fortran
;  routine.
;-
;
 on_error,2              ;Return to caller if an error occurs
 m = N_elements(v) ;# of input pnts
 if N_params(0) eq 2 then begin ;Regular?
  r = findgen(x)*(m-1)/(x-1>1) ;Grid points in V
  rl = long(r)  ;Cvt to integer
  dif = v(1:*)-v  ;V(i+1)-v(i)
  return, V(rl) + (r-rl)*dif(rl) ;interpolate
  endif
;
 if n_elements(x) ne m then $
  message,'V and X must have same # of elements'
 n= n_elements(u) ;# of output points
;
;  If UNIX, then the routine name will have the form "_name" instead of "NAME".
;
 ROUTINE_NAME = 'INTERPOL'
 IF !VERSION.OS NE "vms" THEN ROUTINE_NAME = "_" + $
  STRLOWCASE(ROUTINE_NAME) + "_c"
;
;  Form the name of the sharable object file.
;
 IF !VERSION.OS EQ "vms" THEN FILENAME = "SERTS_EXTERNAL" ELSE BEGIN
  FILENAME = GETENV("SERTS_EXTERNAL")
  IF FILENAME EQ "" THEN MESSAGE,  $
   "Environment variable SERTS_EXTERNAL not found."
 ENDELSE
;
 R = FLOAT(0*U)
 TEST =  CALL_EXTERNAL(FILENAME,ROUTINE_NAME,FLOAT(X),FLOAT(V),LONG(M ), $
```

```
  FLOAT(U),R,LONG(N))
;
 return,r
 end
```