
Subject: Re: Hankel Transform in IDL
Posted by [thompson](#) on Tue, 18 Jun 2002 15:02:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

jl0477@hotmail.com (Jerome LOUIS) writes:

> I am looking for a routine to compute Hankel transforms in IDL,
> and I wonder if someone have it and could send it to me..

> Thanks

> J*r*me

> ps: I saw on this group that Georg Pabst had this program one year
> ago. I tried but could not reach him...

Here's one I wrote a long time ago.

Bill Thompson

```
FUNCTION HANKEL,F
;
; This function returns the Hankel transform of the argument.
;
S = SIZE(F)
IF S(0) NE 1 THEN BEGIN
  PRINT, '*** Variable must be a one-dimensional array, name= F, routine HANKEL.'
  RETURN,F
ENDIF
;
X = INDGEN(F)
K = ( 2. * !PI / FLOAT(N_ELEMENTS(X)) ) * X
SC = 0.*X + 1.
IF N_ELEMENTS(SC) GT 3 THEN BEGIN
  SC(0) = 3.D0 / 8.D0
  SC(1) = 7.D0 / 6.D0
  SC(2) = 23.D0 / 24.D0
ENDIF
;
H = BES0( K # X ) # ( K * F * SC )
;
RETURN,H
END
```

Subject: Re: Hankel Transform in IDL

Posted by [thompson](#) on Tue, 18 Jun 2002 15:19:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

thompson@orpheus.nascom.nasa.gov (William Thompson) writes:

> jl0477@hotmail.com (Jerome LOUIS) writes:

>> I am looking for a routine to compute Hankel transforms in IDL,
>> and I wonder if someone have it and could send it to me..

>> Thanks

>> J*r*me

>> ps: I saw on this group that Georg Pabst had this program one year
>> ago. I tried but could not reach him...

> Here's one I wrote a long time ago.

I guess that really is old! It still has the command

```
X = INDGEN(F)
```

which used to work in IDL's pre-history days!

Here's a corrected version, plus the routine bes0.pro which it calls.

Bill Thompson

```
FUNCTION HANKEL,F
;
; This function returns the Hankel transform of the argument.
;
S = SIZE(F)
IF S(0) NE 1 THEN BEGIN
  PRINT, '*** Variable must be a one-dimensional array, name= F, routine HANKEL.'
  RETURN,F
ENDIF
;
X = DINDGEN(N_ELEMENTS(F))
K = ( 2.D0 * !DPI / DOUBLE(N_ELEMENTS(X)) ) * X
SC = 0.D0*X + 1.D0
IF N_ELEMENTS(SC) GT 3 THEN BEGIN
  SC(0) = 3.D0 / 8.D0
  SC(1) = 7.D0 / 6.D0
  SC(2) = 23.D0 / 24.D0
```

ENDIF

;
H = BES0(K # X) # (K * F * SC)

;
RETURN,H
END

FUNCTION BES0,X_ARRAY

;
A = [-2.2499997D0, +1.2656208D0, -0.3163866D0, +0.0444479D0, \$
 -0.0039444D0, +0.0002100D0]
B = [+0.79788456D0, -0.00000077D0, -0.00552740D0, -0.00009512D0, \$
 +0.00137237D0, -0.00072805D0, +0.00014476D0]
C = [-0.78539816D0, -0.04166397D0, -0.00003954D0, +0.00262573D0, \$
 -0.00054125D0, -0.00029333D0, +0.00013558D0]

;
X = X_ARRAY
IF N_ELEMENTS(X) EQ 1 THEN X = [X,0]
B0 = 0.*X

;
W = WHERE(X EQ 0)
IF W(0) NE -1 THEN B0(W) = 1

;
W = WHERE((X GT 0) AND (X LE 3))
IF W(0) NE -1 THEN BEGIN
 XX = (X(W)/3.D0)^2
 B0(W) = 1 + A(0)*XX
 Q = XX
 FOR I = 1,5 DO BEGIN
 Q = Q*XX
 B0(W) = B0(W) + A(I)*Q
 ENDFOR
ENDIF

;
W = WHERE(X GT 3)
IF W(0) NE -1 THEN BEGIN
 XX = 3.D0 / X(W)
 Q = 1.D0
 F0 = B(0)
 FOR I = 1,6 DO BEGIN
 Q = Q*XX
 F0 = F0 + B(I)*Q
 ENDFOR

;
THETA = X(W) + C(0)
Q = 1.D0
FOR I = 1,6 DO BEGIN

```
Q = Q*XX
THETA = THETA + C(I)*Q
ENDFOR
B0(W) = F0 * COS(THETA) / SQRT(X(W))
ENDIF
;
IF N_ELEMENTS(X_ARRAY) EQ 1 THEN B0 = B0(0)
RETURN,B0
END
```
