
Subject: Variable Type conversions and ATAN function
Posted by [Jason P. Meyers](#) on Mon, 06 Nov 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I recently discovered an interesting curiosity while trying to debug one of my homework problems. It turns out the problem was partly my fault (that's now fixed) and partly IDL's as best I can figure out. Below, you will find part of a journal file I was working with which demonstrates the curiosity I discovered. The curiosity is that sometimes $\text{ATAN}(0.0,0.0) = 0.0$ and other times $\text{ATAN}(0.0,0.0) = 3.14159$. I can't figure out what causes the ATAN to return 0.0 vs 3.14159. In the following example, the third element of FK1 vs FK2 is a good example. I have tried this with different fn vectors and get similar results. Any and all insight would be greatly appreciated.

Thanks,
Jason Meyers
Ph.D. Student, Center for Imaging Science
Rochester Institute of Technology
jpm7934@rit.edu

Program listing:

```
; IDL Version 5.3.1 (Win32 x86)
; Journal File for jason@BLACKBIRD
; Working directory: E:\Jason\Classes\Dip\idlhw\hw6
; Date: Fri Nov 03 12:33:29 2000

; Create a simple vector
fn=float([1,1,1,1,1,0,0,0,0])
; Take its FFT
fk=fft(fn)

; Here is where the fun begins. I am attempting to set a zero threshold
value
; for the complex vector fk.
;
; NOTE: The following line of code should produce the same output as
the
;   for loop. However, it does not. The difference is not
detectable
;   until the FK array is passed to the ATAN function later on.
fk1=fk*complex((abs(fk) gt 1e-4),0.0)
; make a copy
fk2=fk
```

```
for ii = 0,N_elements(fn)-1 do if abs(fk2[ii]) lt 1.0e-4 then
fk2[ii]=complex(0,0)
```

```
; Calculate the phase of each element in both vectors.
```

```
fkph1=atan(imaginary(fk1),float(fk1))
```

```
fkph2=atan(imaginary(fk2),float(fk2))
```

```
; Print the various arrays
```

```
print,'FK:'
```

```
print,fk
```

```
print,'FK1:'
```

```
print,fk1
```

```
print,'FK2:'
```

```
print,fk2
```

```
print,'Phase of FK1:'
```

```
print,fkph1
```

```
print,'Phase of FK2:'
```

```
print,fkph2
```

```
print,'Difference: FK1 - FK2:'
```

```
print,fk1 - fk2
```

IDL Output:

```
IDL> @test
```

```
FK:
```

```
( 0.500000, 0.000000)( 0.100000,
-0.307768)(-3.72529e-009, 0.000000)( 0.100000, -0.0726543)(
7.45058e-009, 0.000000)
```

```
( 0.100000, 0.000000)( 7.45058e-009, 0.000000)(
0.100000, 0.0726543)(-3.72529e-009, 0.000000)( 0.100000,
0.307768)
```

```
FK1:
```

```
( 0.500000, 0.000000)( 0.100000, -0.307768)(
0.000000, 0.000000)( 0.100000, -0.0726543)( 0.000000,
0.000000)
```

```
( 0.100000, 0.000000)( 0.000000, 0.000000)(
0.100000, 0.0726543)( 0.000000, 0.000000)( 0.100000,
0.307768)
```

```
FK2:
```

```
( 0.500000, 0.000000)( 0.100000, -0.307768)(
0.000000, 0.000000)( 0.100000, -0.0726543)( 0.000000,
0.000000)
```

```
( 0.100000, 0.000000)( 0.000000, 0.000000)(
0.100000, 0.0726543)( 0.000000, 0.000000)( 0.100000,
0.307768)
```

```
Phase of FK1:
```

```
0.000000 -1.25664 3.14159 -0.628318 0.000000
0.000000 0.000000 0.628318 0.000000 1.25664
```

Phase of FK2:

```
0.000000 -1.25664 0.000000 -0.628318 0.000000
0.000000 0.000000 0.628318 0.000000 1.25664
```

Difference: FK1 - FK2:

```
( 0.000000, 0.000000)( 0.000000, 0.000000)(
0.000000, 0.000000)( 0.000000, 0.000000)( 0.000000,
0.000000)
( 0.000000, 0.000000)( 0.000000, 0.000000)(
0.000000, 0.000000)( 0.000000, 0.000000)( 0.000000,
0.000000)
```
