Subject: Different FFT-results on vector and array?!
Posted by jan on Tue, 30 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

Distribution: world

When checking out amplitude and phase of a given frequency in several subwindows of a given timewindow I ran into the following problem:

I have to FFT the array "data(dim,3)" where dim is a given long integer. In the first step I perform the FFT on the whole array at once and in the second try I perfom the FFTs on data(*,0), data(*,1) and data(*,2).
The results are stored in both cases in temp(*,*).
My problem is: these two strategies yield different results...

Following I attached the relevant code of both cases...
(variables n1, n2, k, freq and delta are predefined in the code before)

---FFT on the array at once:

```
REPEAT BEGIN
  dim   = LONG(N_ELEMENTS(data(n1:n2,0)))
  ff    = FLTARR(dim/2+1,3)
  pp    = FLTARR(dim/2+1,3)
  temp  = COMPLEXARR(n2-n1+1,3)
    temp2 = 1000.*data(n1:n2,*)
    temp  = FFT(temp2,-1)
  ff = FLOAT(temp(0:dim/2,*))
  pp = IMAGINARY(temp(0:dim/2,*))
  k = k + 1
  a(k,freq,*) = ff(freq,*)  ;a is FLTARR(k_max,freq_max,3)
  p(k,freq,*) = pp(freq,*)  ;p is FLTARR(k_max,freq_max,3)
  n1 = n1 + delta
  n2 = n2 + delta
ENDREP UNTIL (<certain condition is true>)
```

---FFT on vectors containing the array-columns:

```
REPEAT BEGIN
  dim   = LONG(N_ELEMENTS(data(n1:n2,0)))
  ff    = FLTARR(dim/2+1,3)
  pp    = FLTARR(dim/2+1,3)
  temp  = COMPLEXARR(n2-n1+1,3)
    temp2 = 1000.*data(n1:n2,0)
    temp(*,0)  = FFT(temp2,-1)
    temp2 = 1000.*data(n1:n2,1)
    temp(*,1)  = FFT(temp2,-1)
    temp2 = 1000.*data(n1:n2,2)
```

```
   temp(*,2)  = FFT(temp2,-1)
  ff = FLOAT(temp(0:dim/2,*))
  pp = IMAGINARY(temp(0:dim/2,*))
  k = k + 1
  a(k,freq,*) = ff(freq,*)  ;a is FLTARR(k_max,freq_max,3)
  p(k,freq,*) = pp(freq,*)  ;p is FLTARR(k_max,freq_max,3)
  n1 = n1 + delta
  n2 = n2 + delta
ENDREP UNTIL (<certain condition is true>)
---
```

When plotting a(i,freq,[0,1,2]) over i I get different results for the above two cases. All surrounding code, including the stop-repeat-condition is _exactly_ the same for both routines. I'm sure there must be a pretty silly mistake - however, I can't figure out where it is ...

Any hints?
Thanks in advance,
Jan
(e-mail is welcome, too)

Jan J. Krueder
German Aerospace Research Establishment
Institute of Space Simulation
privat : jan@spacelab.greenie.muc.de
office : jan.kruder@europa.rs.kp.dlr.de
