

Appendix F: Scaling of Image Data

With reference to chapter three, it is required to compare the *shape* of the object phase, stored as OBJ(K), with the shape of the image intensity, stored as IMAGE(K). The mean of the OBJ(K) is zero, the function being a superposition of cosines each of mean zero, but the image intensity has a dc bias signal. Thus this signal is subtracted from IMAGE(K) so that both functions have zero mean. Assuming no shift of one function relative to the other has occurred, the only difference between OBJ(K) and IMAGE(K) allowed is a difference of scale. A scaling parameter is found from a least squares analysis assuming the relation

$$\text{Object} = \text{SCALE} \times \text{Image} \quad (1)$$

so that SCALE is less than one. (Note that the image data is *reflected* by a FORTRAN subroutine to counter the effect of reflection about the Y-axis which occurs after re-transformation of the spectrum.) Before any cross-correlation etc. is performed the image data IMAGE(K) is then replaced by SCALE \times IMAGE(K). It is therefore acceptable that the value of the cross-correlation may slightly exceed unity in value due to this scaling operation.

Elimination of Phase Errors

The relation between wavelength in the object plane and the pixel to which the associated frequency of that wave is mapped to in the frequency plane is as follows. If one complete cycle of the object lies on P+1 pixels maximum to maximum, then the wavelength of the object is actually P pixels long. For the case of a 256 point Fourier Transform, such a wavelength is mapped to pixels Pix(P) where

$$\text{Pix}(P) = 129 \pm \frac{256}{P} \quad (2)$$

Thus in the Fourier series representation of the object the fundamental frequency may, as is common, be selected so as to be mapped to the first pixel either side of the zero frequency pixel (129) in the frequency plane. From the above conditions, it must have a wavelength which is 256 pixel separations long and so lie on 257 pixels. This can be realised by the function

$$a_1 \cos\left(2\pi \frac{(K-1)}{257} + \Phi_n\right) \quad (3)$$

where K is the object pixel coordinate and $1 < K < 256$.

It is necessary to perform this operation in order to recover the phase information Φ_n unaltered in the frequency plane. If this precaution is not taken no comparison can be made between the phase Φ_n of the Fourier component and the measured phase of the spectrum as determined from the real and imaginary parts of the Fourier Transform.

Accurate Recovery of Phase

Given that the precautions above have been taken, there remains one further slight problem with phase recovery. The phase of the spectrum at each frequency position is expected to be $i e^{i\Phi_n}$. Thus before a comparison can be made between ideal phase and measured phase the amount $[(\pi)/2]$ must be subtracted from the measured phase. Complications arise because the measured phase, as determined by the FORTRAN intrinsic function ATAN2, lies in the range $-\pi < \theta < +\pi$. Consider the case where the object phase $\Phi_n = 0.49\pi$. The measured phase in the spectrum would ideally be this value plus $[(\pi)/2]$, which is still less than $+\pi$ and so should present no problem to the ATAN2 function.

Chapter three shows that phase errors always occur however, so that the measured phase θ is more likely to be

$$\theta = \Phi_n \pm \delta + \frac{\pi}{2} \quad (4)$$

and the inclusion of δ can throw the measured phase over the limit of $+\pi$. The ATAN2 function would recover θ as a negative phase, and subtraction of $[(\pi)/2]$ would further compound this problem. Figure 6.1 illustrates the problem.

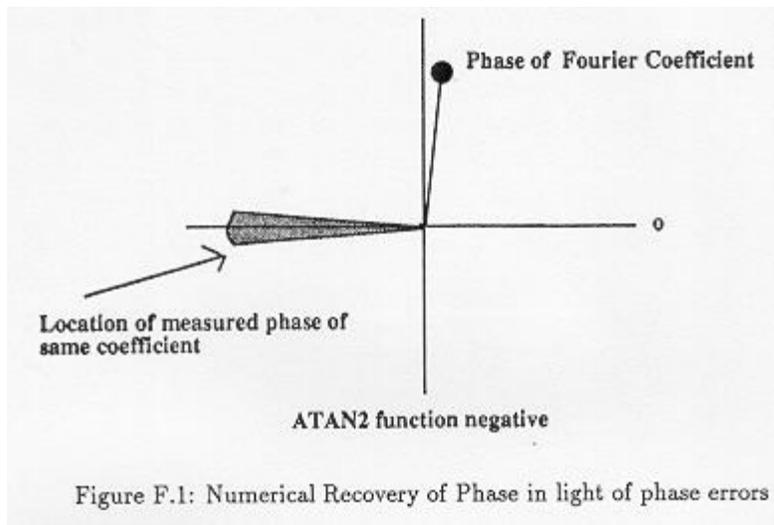


Figure F.1: Numerical Recovery of Phase in light of phase errors

It is thus necessary to include an extensive series of tests to check whether such an event has occurred, comparing recovered phase with the quadrant of Φ_n etc. before the difference between the two values can be made. The tests made in the actual simulation are detailed below.

'K' denotes which of the 50 random spectrums is being used and
 'J' denotes which freq. in that spectrum we are on.

```

DO 200 K=1,50 !.....!Select spectrum:
PHIDEV(K)=0.0 !.....!Phase deviation for K'th
!spectrum as a whole.
DO 100 J=1,NC !.....!Cycles through frequencies.

BUGG=PHI(J)/PI !.....! Actual phase of coefficient.
RUGG=(NEWPHA(J)/PI) !.....! Measured phase via ATAN2.
T1=0
T2=0
IF (BUGG.GE.0.AND.BUGG.LT.(0.5)) THEN
  T1=1 !.....!Phase in 1st quadrant
ELSE IF (BUGG.GE.(0.5)) THEN
  T1=2 !.....!Phase in 2nd quadrant
ENDIF

IF (T1.EQ.1.AND.RUGG.GT.0) THEN!...!DRUGG is actual difference
  DRUGG=RUGG-0.5 !of phase after ATAN2 effects
ELSE IF (T1.EQ.1.AND.RUGG.LT.0) THEN !are accounted for.
  DRUGG=RUGG+1.5
ELSE IF (T1.EQ.2.AND.RUGG.LT.0) THEN
  DRUGG=RUGG+1.5
ELSE IF (T1.EQ.2.AND.RUGG.GT.0) THEN
  DRUGG=RUGG-0.5
ELSE IF (BUGG.LT.0) THEN
  DRUGG=RUGG-0.5
ENDIF

Y=ABS(DRUGG-BUGG) !.....! Take modulus of phase error.
IF (Y.GE.1) THEN
  Y=ABS(Y-2) !.....!A catch-all, never occurred in
ENDIF !any simulation once.

PHIDEV(K)=PHIDEV(K)+Y !.....!Increment phase error for this
!spectrum.

```

```

100 CONTINUE
PHIDEV(K)=PHIDEV(K)/(1.0*NC)
200 CONTINUE

```

File translated from T_EX by [T_TH](#), version 3.02.

On 27 Oct 2001, 23:43.