HESSI Variables and Equations used for Image Reconstruction

- t_i= **time** = Binary microseconds from reference start time
- N_i= **counts** = Number of events per time bin
- x_i^A, y_i^A= **x_asp, y_asp** = Yaw, pitch offset of sun-center from imager axis in spacecraft coordinates for the ith time bin
- X_m, Y_m= **X_pixoff, Y_pixoff** = Pixel m offset from map center in inertial coordinates
- X^M, Y^M= **X_map, Y_map** = Map center offset from Sun center in inertial coordinates (user input)
- x^M, y^M= **x_map_sc, y_map_sc** = Map center offset from imager axis in S/C coords

  x_map_sc = x_asp + X_map*cos(roll_angle) - Y_map*sin(roll_angle)

  y_map_sc = y_asp + Y_map*cos(roll_angle) + X_map*sin(roll_angle)

  $$x^M = x^A + X^M cos(\alpha) - Y^M sin(\alpha)$$
  $$y^M = y^A + Y^M cos(\alpha) + X^M sin(\alpha)$$

- α= **roll_angle** = Roll angle (radians), of spacecraft X-axis (measured clockwise from celestial N)
- β= **grid_angle** = Grid slit orientation in spacecraft coordinates (m radians) (from parameter file)
- τ= **livetime** = Binary fraction giving duty cycle of detector
- T= **gridtran** = Grid-response-matrix transmission
- A_h= **modamp** = Grid-response-matrix amplitude
- Q_h= **peak_resp_offset** = Peak grid-response-matrix offset (from hsi_grm–Step 5)
- p= **ang_pitch** = Angular pitch of subcollimator grids (from parameter file)
- Θ_i= **phase_map_ctr** = Phase of map center relative to line of maximum response (Step 6)

  phase_map_ctr = 2*!pi*h*(x_map_sc*cos(grid_angle) + y_map_sc*sin(grid_angle) + peak_resp_offset)/ang_pitch

  $$\Theta_i = 2\pi h(x^M cos(\beta) + y^M * sin(\beta) + Q_h)/p$$

- Λ= **phase_pixel** = Phase of pixel m relative to phase of map center

= 2*!pi*h*(X_pixoff[m]*cos(roll_angle-grid_angle) - Y_pixoff[m]*sin(roll_angle-grid_angle)) / ang_pitch

$\Lambda = 2\pi h(X_m cos(\alpha - \beta) - Y_m sin(\alpha - \beta))/p$

- E_{mi}= **counts_exp** = Expected counts in time bin i from pixel m:

  = F[m,i]*livetime * gridtran*(1 + modamp*cos(phase_map_ctr + phase_pixel))

  $E_{mi} = F_{mi}\tau T(1 + A_h cos(\Theta_i + \Lambda_m)$

- F= **flux** = Postulated flux

Seven Steps to Calibrated Modulation Profiles

- **STEP 0: INPUT LEVEL-0 PACKETS**

  Select a time range.

  Input level-0 event data packets for the selected time range.

  IDL SNIPPET:

  ```
  t_start=0
  t_end=2^21 ; (binary micro sec for nominal half rotation)
  ```

- **STEP 1: SCORE CREATION**

  Select an energy range and a subcollimator.

  Form a score, listing individual events satisfying these criteria.

  Attribute of each event: Time 4-bytes (64 bus resolution)

  IDL SNIPPET:

  ```
  energy_edges=[50.,100.] ; ( keV)
  atten_state=0
  det_index=8 ; (0–8)
  ang_pitch=348.30 ; (asec–to be converted to microrad)
  ```

hsi_packet2score,packet,score ; TBD as of 5/99

- **STEP 2: TIME-BINNING**

  Select time-binning parameterization.

  Group events in the score into time-bins with predetermined widths.

  (Special case: Each time-bin = 64 bμs.)

  There should be one entry per time bin, EVEN IF NO EVENTS

  Attribute of each time-bin.

  Time: 4-bytes (64 bμs resolution)

  counts: Number of events 4-bytes

  IDL SNIPPET:

  ```
  bin_length=512 ; ( binary microsec)
  counts=hsi_score2hist( score, icoll,bin_length=bin_length)
  counts=histogram(time,min=t_start,max=t_end,binsize=bin_length)
  bin_times=bin_length/2+bin_length*lindgen(n_elements(counts))
  ```

- **STEP 3: ASPECT ASSOCIATION**

  Associate an aspect solution with each time-bin.

  time: Time 4-bytes (64 bμs resolution)

  counts: Number of events 4-bytes (counts)

  y_asp: Pitch offset 2-bytes (microradians)

  x_asp: Yaw offset 2-bytes (microradians)

  roll_angle Roll angle 4-bytes (microradians)

  IDL SNIPPET:

  ```
  aspect = Hsi_aspect_solution( time )
  x_asp=aspect.x_asp
  y_asp=aspect.y_asp
  roll_angle=aspect.roll_angle
  xy_interpolated=roll_interpolate(x_asp,y_asp, etc)
  ```

- **STEP 4: LIVE-TIME ASSOCIATION**

  Associate a live-time measure with each time-bin.

  The actual time of the time-bin can be dropped. It is no longer needed and in any case, can be reconstructed.

  Attribute of each time bin:

  counts: Number of events 4-bytes
  livetime: Live-time measure 2-bytes (binary fraction)
  y_asp: Pitch offset 2-bytes (microradians)
  x_asp: Yaw offset 2-bytes (microradians)
  roll_angle Roll angle 4-bytes (microradians)

  IDL snippet:

  hsi_livetime_sim, score, deadtime_sim_params,livet_ctr_str

- **STEP 5: ASSOCIATE GRID CALIBRATION**

  Select a map center,relative to sun center. Select desired harmonic of grid response. Determine the grid response matrix parameters associated with each time bin. (This is assumed to include source-location independent factors such as detector efficiency.)

  Attribute of each time bin:

  counts: Number of events 4-bytes
  livetime: Live-time measure 2-bytes (binary fraction)
  y_asp: Pitch offset 2-bytes (microradians)
  x_asp: Yaw offset 2-bytes (microradians)
  roll_angle: Roll 4-bytes (microradians)
  grm.modphz: Modulation phase 2-bytes (TBD)

  IDL snippet:

  x_map=600.
  y_map=200. ; select a map center close to flare
  ; Map-center in spacecraft coords: (Sun spins CCW as roll_angle increases.)

```
x_map_sc=x_asp+X_map*cos(roll_angle)-Y_map*sin(roll_angle) ; asec
y_map_sc=y_asp+Y_map*cos(roll_angle)+X_map*sin(roll_angle) ; asec
R_map_sc=sqrt(x_map_sc^2+y_map_sc^2)/3600 ; convert to degrees
azimuth_map_sc=atan(y_map_sc,x_map_sc)
offax_position=fltarr(2,n_elements(roll_angle))
gridtran=0*roll_angle
modamp=gridtran
peak_resp_offset=gridtran
h=1 ; We'll do only the fundamental
for i=0L,n_elements(roll_angle)-1 do begin
  offax_position=[R_map_sc[i],azimuth_map_sc[i]]
  hessi_grm, energy_edges, det_index, grm, atten_state, offax_position
  gridtran[i] = (grm.gridtran)[h-1]
  modamp[i] = (grm.modampl)[h-1]
  peak_resp_offset[i]=(grm.modphz)[h-1] ; presumably has units of ang_pitch but hessi_grm.pro doesn't say yet
endfor
; peak_resp_offset will be a constant vector until Gordon updates hessi_grm
```

- **STEP 6: DETERMINE PHASE CALIBRATION**

  Combine map center, modulation phase, pitch and yaw offsets into a phase offset.

  counts: Number of events   4-bytes
  livetime: Live-time measure   2-bytes (binary fraction)
  roll_angle: Roll angle   4-bytes (microradians)
  grm.modphz: Phase offset   2-bytes (TBD)

  IDL snippet:

```
nx=64L
npixels=nx^2
pxl_size=1.0
```

```
; square map of size nx X nx: (Map center is between pixels if
    nx is even)
x_pixoff=pxl_size*reform((findgen(nx)-(nx-1.)/2.)#replicate(1,nx),npixels)
y_pixoff=pxl_size*reform(replicate(1.,nx)#(findgen(nx)-(nx-1.)/2.),npixels)
xasp=x_asp[iroll] & yasp=y_asp[iroll]
rollangl=roll_angle[iroll] ; S/C rotates CW against sun as roll
    increases
phase_map_ctr=2*!pi*h*(x_map_sc*cos(grid_angle)+y_map_sc*sin(grid_angle)
    + peak_resp_offset)/ang_pitch
etc
```