Modifications to the Computation of NEOWISE-R Coverage Progress

WSDS D-T048

I. Introduction

The Perl scripts covprogress and survprogress are used in NEOWISE-R Operations to compute and plot the cumulative coverage in units of framesets per Hierarchical Equal-Area iso-Latitude Pixel (HEALPix) of the celestial sphere as the mission progresses. Recently these scripts have exhibited defects as a result of: (i) the first "roll-over" of the numeric part of scan ID in December 2018, when suffixes r and s were introduced and the numeric part was reset from 99979 for a and b scans to 01000 for r and s scans; (ii) the use of the scripts on a monthly basis, even though they were hardcoded for use on a weekly basis, starting in August 2021; (iii) the second "roll-over" of the numeric part of scan ID in late October, 2022, when the numeric part of scan ID of r and s scans overtook that for the first a and b scans of the mission, at 44212. The last of these effects rendered the scripts unusable because the nearly 9-year accumulation of the NEOWISE-R mission was reduced to approximately 3 weeks. This document describes the changes that have been developed from November 2022 to February 2023 to recover the functionality of the scripts in order to produce correct cumulative coverage for each HEALPixel.

II. Script covprogress

The effects enumerated in Section I, above, were listed in the chronological order of the NEOWISE-R survey observations to which they applied. Heretofore we describe the development effort in the order in which it was implemented in the Perl scripts.

II.1 Roll-over of numeric portion of scan ID where r- and s-suffixed scans reached 44212 and above, overtaking a- and b-suffixed early NEOWISE-R mission scans

When coverage progress is computed on a particular ending date given as a parameter, the script covprogress determines, via a query to the frame index, the scan range observed within the interval of a number of whole days prior to and up through UTC noon on the ending date. The number of days was fixed at 7 prior to the current implementations, as described in Section II.2, but now it is determined by the new parameter *scan_interval* (Section II.2). As part of its computations, covprogress used to search for the previous scan range when the script was run last time, by using an alphanumeric sorting of previously observed scan IDs. The problem with this approach is that when the "second roll-over" of numeric scan ID occurred in 2022 late-Oct, wherein the most recent r- and s-suffixed scan IDs had numeric portions reaching and then exceeding the value of 44212 corresponding to the first scan of the NEOWISE-R mission, the search for a preceding scan range using alphanumeric sorting of scan ID failed.

The last time that covprogress ran relatively well in the above respect was for an ending date of 2022 Oct 10. The scan range determined from the frame index was 43624r-43839r. The initial scan in this range was observed 7 days prior to the ending date. This and similar scan ranges for other ending dates gave us the first indication that the interval for the scan ranges was always fixed at 7 days, prompting us to investigate and eventually modify the code to allow for other intervals such as 30 days (Section II.2). Each time that covprogress runs, for a given ending date, a so-called HPIC (HEALPix) FITS file is generated. For this run, the generated HPIC file is **cov_progress-hpic-221010T120000Z-43624r_43839r-ecl-all.fits**. The previous instance of coverage progress is searched by the script, in order to add the current coverage accumulation to the previous one. The Operations directory is searched for all other HPIC files. The alphanumeric sorting of scan IDs determined that the previous instance of HPIC, designated as the "**last HPIC**," was **cov_progress-hpic-220913T120000Z-42796r_43010r-ecl-all.fits**. As the name implies, it was computed on 2022 Sep 13, which was a correct inference of the closest (in time) preceding instance of running the script.

Following the above coverage progress run, the script was run for an ending date of 2022 Nov 16. The scan range determined from the frame index was 44760r-44896r, which is a 7-day interval. The generated HPIC file is **cov_progress-hpic-221116T120000Z-44760r_44896r-ecl-all.fits**. The alphanumeric sorting of scan IDs determined that the previous instance of HPIC, or so-called "**last HPIC**," was **cov_progress-hpic-131231T120000Z-44537b_44749b-ecl-all.fits**. As the name of the HPIC file implies, it corresponds to the coverage ending on 2013 Dec 31, and is grossly in error as the "most recently preceding" or "last" HPIC file before the current scan range. That is, the use of alphanumeric sorting of scan IDs, coupled with the "second roll-over," led to the gross misidentification. The scan range in this erroneous HPIC file is 44537b-44749b (approximately the third week of the NEOWISE-R mission).

The solution to this problem was to introduce, in the covprogress script, subroutines *scan_order* and *in_scan_order*, which sort scan IDs alphanumerically but taking into account that a- and b-suffixed scans precede r- and s-suffixed scans in the NEOWISE-R mission. Other suffixes (t, u, v, w, y) are also considered as later than a and b suffixes in the mission. The subroutine *in_scan_order* is a particular case that compares two scan IDs and returns a Boolean value of *true* if the first scan ID is less than or equal to the second, and *false* otherwise.

II.2. New parameter in covprogress to compute accumulated coverage during an arbitrary number of days

It is recognized that coverage progress may be computed by the Operations team on a weekly basis (as was the case from the beginning of the NEOWISE-R mission and up through 2021 Aug 3), then roughly on a monthly basis through the present, but with actual durations of between 25 and 40 days. Previous to the implementations described here, the actual coverage period for accumulation used by the script covprogress was fixed (hardcoded) at 7 days. The result is that the cumulative coverage up through 2022 Nov 16 would be underestimated by the following factor: with correctly computed coverage accumulation per month computed during 92 months through 2021 mid-Aug, and 14 months of incorrectly computed coverage at only 25% (7 days out of every month) from then through 2022 Nov 16, the underestimate of cumulative coverage is: $1 - (14 \bullet 0.25 + 92) / (14 + 92) = \sim 10\%$, such as at the ecliptic poles.

The solution to this underestimate of coverage was to introduce a new parameter in the call of covprogress, indicating the number of days from the end of the previous computation of coverage to the ending date of the current coverage. This parameter is called *scan_interval* and its units are calendar days. The user is advised to use values of *scan_interval* between 7 and 40 days, inclusive. Longer intervals may cause the script to hang for inordinate amounts of time. In order to correct the coverage accumulation from 2021 Aug 3 through 2022 Nov 16, we re-run coverage covprogress in sequences of approximately monthly intervals, as done before but now with the use of *scan_interval*. Table 1 shows the parameters of these re-runs:

Table 1. Coverage progress re-run parameters

ENDING COVERAGE CALENDAR DATE [YYMMDD]	ENDIND COVERAGE MJD [days]	SCAN_INTERVAL [days]
210913	59469.5	40
211011	59498.5	29
211108	59526.5	28
211213	59561.5	35
220117	59596.5	35
220214	59624.5	28
220315	59653.5	29
220412	59681.5	28
220517	59716.5	35
220615	59745.5	29
220712	59772.5	27
220817	59808.5	36
220913	59835.5	27
221010	59862.5	27
221116	59899.5	37

II.3 Roll-over of numeric portion of scan ID where a- and b-suffixed scans reached a value of 99979a and were subsequently followed by r- and s- scans starting at 01000r

On 2018 Dec 10, the numeric portion of scan ID transitioned from the highest number in a- and b-suffixed scans, namely 99979a, to a starting low number for new r- and s-suffixed scans, namely 01000r. Shortly after this so-called "first roll-over" of numeric scan ID, errors in the annotation of all-sky coverage progress plots arose. For coverage progress with an ending date of 2018 Dec 18, the sky coverage ordinal number became "1" (instead of a true value of 11). Then on 2018 Dec 25, the beginning scan of the mission was listed as "01034r," instead of a true value of 44212a, and this value was propagated to the legends of subsequent coverage progress plots, up through the end of 2022 when we implemented the changes described in this document. The fractional sky coverage also became unreliable since around that time. As a recent example, for coverage progress ending on 2022 May 17, the all-sky plot displays a coverage number of "6," instead of a true value of 11, beginning scan of the mission as "01034r," as already mentioned above, and a coverage fraction of "48%," instead of a true value closer to 83%.

We addressed one of the above problems, namely the legend listing the beginning scan of the mission, through modifications to covprogress. For a given ending date of coverage progress, a scan range is identified through a query to the frame index, and the first scan ID in this range is thus determined. The immediately preceding instance of coverage progress is searched in the Operations directory, and a so-called "**last HPIC**" FITS file is identified, if it exists. If the latter does not exist, then the first scan of the mission is identified as the first scan ID in the scan interval of the current coverage progress run. Otherwise, if a last-HPIC FITS file exists, then the FITS header keyword 'LSTHPSC1' is read and its value is compared with the first scan ID in the scan interval of the current run. The "smallest" of these two scan IDs is determined to be the first scan of the mission. On 2022 Dec 18, the current HPIC FITS file was **cov_progress-hpic-181218T120000Z-01034r_01245r-ecl-all.fits**, and the first scan of the mission was correctly listed as '44212a'. This HPIC file became the "last HPIC" file of the subsequent run of coverage progress on 2022 Dec 25. The values of the keywords 'LSTHPSC1' and 'SCAN1' in this HPIC file had the value '01034r'. In the 2022 Dec 25 run, the comparison of 'LSTHPSC1' and '44212a' resulted in '01034r' < '44212a' and consequently '01034r' became the first scan of the mission from then on. Instead of a simple alphanumeric comparison of scan IDs, the subroutine *in_scan_order* was invoked for the two arguments 'LSTHPSC1' and the first scan ID in the scan interval of the current coverage progress run. In addition, in order to avoid having to re-run a large number of monthly coverage progress instances, the preceding last HPIC FITS file was altered in the header keyword 'LSTHPSC1' to have a value of '44212a' for the above listed last HPIC of the 2018 Dec 25 run. The script covprogress correctly annotated the first scan of the mission as 44212a for this run. This process was repeated for a more recent "last HPIC" file, namely for the 2023 Jan 16 run, where the last HPIC file is **cov_progress-hpic-221213T120000Z-44974r_45803r-ecl-all.fits**. The 'LSTHPSC1' keyword was similarly edited, and the subroutine *in_scan_order* within covprogress correctly identifies that 44212a < 44974r. The first scan of the mission will thus be listed as 44212a indefinitely.

We attempted to fix the problems of erroneous coverage number and coverage fraction, which are the main outputs of the script survprogress, and which we thought were related to the "first roll-over" of numeric scan ID. However, due to non-existent documentation of this script, very intricate programming style, and its non-functionality, we decided to re-write from scratch a new subroutine called new_survprogress, described in the next Section. As described in Section III, a few elements of survprogress were adopted in the development of new_survprogress.

III. Script new_survprogress

This subroutine, called by covprogress, is meant to compute the ordinal number of sky coverage and the fractional sky coverage, given input MJD and time interval in units of days. These two quantities are then written in the annotation of coverage progress (actually cumulative coverage) all-sky plots and ASCII log files.

Fig. 1 shows one of the coverage progress output plots, which actually represents the cumulative coverage for the NEOWISE-R mission, for an ending date of 2023 Jan 16. An interval of 34 days was chosen to add to the accumulation, from the previously run coverage progress on 2022 Dec 13. This plot has correct values and annotations, after the implementations in this document.

Next are listed the specifications for the development in Perl of this subroutine.

III.1 Algorithm specification of script new_survprogress

The variable $scan_interval (integer) from the main program covprogress should be passed as a command-line parameter in new_survprogress to populate the variable $recent (integer) (number of days = number of degrees).

Perform an initial calculation of $epoch = int(($mjd1 – $mjd0) / 182.5) + 1 where "int" is the typical integer-by-truncation function.

Then the coverage obtained during the $recent number of days interval will be added. This coverage will be in two lunes of opposing ecliptic longitude, shown in projection as two sectors in dark-blue shading in Figs. 2 – 6. Note that the variable $epoch is a scalar (not a 2-element array, unlike in the existing survprogress script), and its numerical definition is also different from that in the existing script.

III.1.i. Scan by scan operations:

From a call to getfix, the script already obtains a list of framesets, with associated ecliptic longitudes $elon and latitudes $elat. **It is very important to discard the "y"-suffix scans from the output of getfix**. For earlier mission getfix queries, "x"-suffix scans are already excluded from the output. Copy the array of longitudes $elon (see Fig. 7 for an example) to a new array $elon_work. The array of longitudes (framesets) should be cleared of those where corresponding latitudes as a function of time switch the sign of their first derivative. These cases are framesets that reach one of the ecliptic poles and the longitudes exhibit a corresponding discontinuity. All framesets from the first that switched sign in their

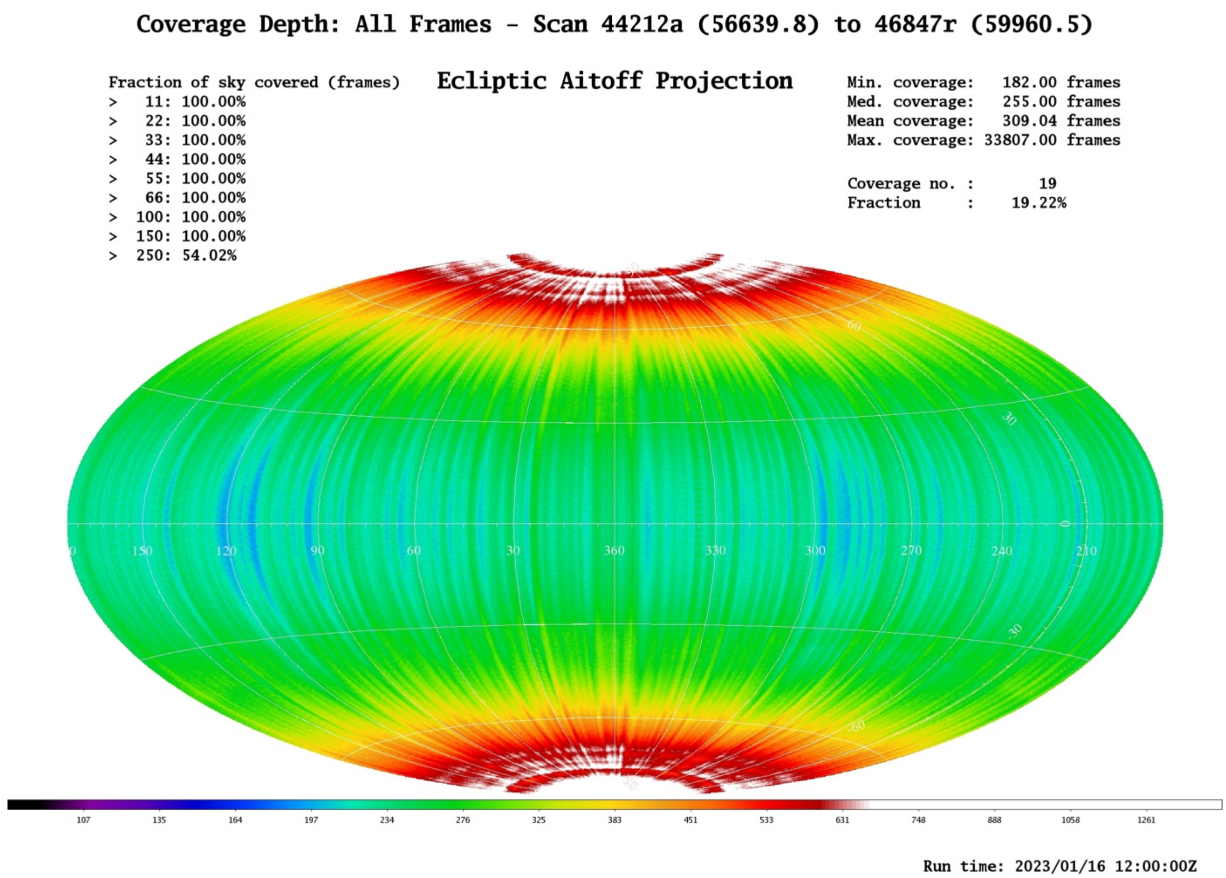

FIG. 1: Output plot from covprogress and new_survprogress, after implementing all algorithmic specifications in this document. The file is called cov_progress-hpic-230116T120000Z-45803r_46847r-ecl-all-aitproj.jpeg and is an Aitoff all-sky projection in ecliptic coordinates of the NEOWISE mission cumulative coverage (framesets per HEALPixel), up through UT noon 2023 Jan 16. The values of coverage and all annotations in this plot are now correct. In particular: the previous run of coverage progress is identified as that of 2022 Dec 13, and the so-called **last HPIC** file is identified by the covprogress script as cov_progress-hpic-221213T120000Z-44974r_45803r-ecl-all.fits (Section II.1); consequently, the accumulation of coverage progress is correctly added to the immediately preceding run of coverage progress. The script covprogress allowed the use of a time interval of 34 days separating the above two runs (Section II.2), thus allowing for continuous accumulation without gaps or overlap. The beginning scan of the mission (44212a) is obtained by covprogress (Section II.3); the ordinal coverage number (19) and the fraction of sky coverage (19.22%) are obtained by new_survprogress (Section III).

first derivative to all those few framesets immediately following at the end of the scan (if the frame is within the 85-percentile of frame number within the scan), or those immediately preceding it at the beginning of the scan (if the frame is within the 15-percentile of frame number within the scan), should be discarded. In addition, framesets whose longitude differs by more than 2 degrees from their immediately preceding or immediately following frameset, and which are at or near the beginning of the scan, or at or near the end of the scan (per the above 15- or 85-percentiles) should be discarded, together with the pertinent neighbor. (see Fig. 8 for an example).

III.1.ii. Operations on the entire array of longitudes

1. Count the number of elements of the array of longitudes and call it N (float).
2. Count the number of longitudes between 0 deg and $recent deg; call it Np (float).
3. Count the number of longitudes > 360 deg – 2 * $recent deg; call it Nn (float).
4. If Np and Nn are each at least 10% of N, then it means that there are a significant number of ecliptic longitudes straddling 0 deg, in which case, for $elon_work > 360 deg – 2 * $recent, do $elon_work = $elon_work – 360 degrees. That is, those longitudes close to (but not beyond) 360 deg, within 2 * $recent (in deg), will be made negative. The rationale for this conversion is that the scans within a lune spanning a number of days = $recent must have continuous values (including small negative $elon_work values that progress to small positive $elon_work values); see Figs. 4 and 5.

Compute $limit_natural = mean($elon_work) and

$limit_complement = $limit_natural – 180 degrees.

The variables $min, $max, and $angle are 2-element arrays; 0-indexing is used in the following expressions. $angle is a dihedral angle.

$min[0] = minimum($elon_work where $limit_complement < $elon_work < $limit_natural)

$max[0] = maximum($elon_work where $limit_complement < $elon_work < $limit_natural)

$angle[0] = $max[0] – $min[0]

Perform 3σ-clipping:

> Compute mean longitude Σ and standard deviation of longitude σ for those longitudes $elon_work[j] that satisfy $min[0] < $elon_work[j] < $max[0] and where j is an index of framesets (can be a large number). Discard from arrays those $elon_work[j] where abs($elon_work[j] –Σ) > 3 * σ (i.e., perform 3σ-clipping).
>
> Get updated $min[0], $max[0], $angle[0] quantities:
>
> $min[0] = minimum($elon_work where $min[0] < $elon_work < $max[0])
>
> $max[0] = maximum($elon_work where $min[0] < $elon_work < $max[0])

(where $min[0] and $max[0] in the r.h.s. of the above equations are the original values, before the update);

$angle[0] = $max[0] – $min[0]

$min[1] = $min[0] + 180 deg; if $min[1] > 360 deg then $min[1] = $min[1] – 360 deg

$max[1] = $max[0] + 180 deg; if $max[1] > 360 deg then $max[1] = $max[1] – 360 deg

Fine-tune $min[1] and $max[1] so they conform better to actual ecliptic longitudes (see Figs. 7, 8):

There are two cases:

(i) If either (or both) Np or Nn are less than 10% of N:
If ($max[0] + 210 deg) < 360 deg then

$min[1] = minimum($elon_work where ($min[1] – 30 deg) < $elon_work <
($max[1] + 30 deg)

$max[1] = maximum(($elon_work where ($min[1] – 30 deg) < $elon_work <
($max[1] + 30 deg)

where it is understood that the values of $min[1] and $max[1] in the r.h.s. of the equations are the original values.

If ($max[0] + 210 deg) > 360 deg then

$min[1] = minimum($elon_work where ($min[0] + 210 deg) < $elon_work < 360 deg

$max[1] = maximum(($elon_work where ($min[1] – 30 deg) < $elon_work <
($max[1] + 30 deg)

(ii) If both Np and Nn are more than 10% of N:
$min[1] = minimum($elon_work where ($min[1] – 30 deg) < $elon_work <
($max[1] + 30 deg)

$max[1] = maximum(($elon_work where ($min[1] – 30 deg) < $elon_work <
($max[1] + 30 deg)

where again it is understood that the values of $min[1] and $max[1] in the r.h.s. of the equations are the original values.

$angle[1] = $max[1] – $min[1]

Perform 3σ-clipping:

Compute mean longitude Σ and standard deviation of longitude σ for those longitudes $elon_work[j] that satisfy $min[1] < $elon_work[j] < $max[1] and where j is an index of framesets (can be a large number). Discard from arrays those $elon_work[j] where abs(elon_work[j] –Σ) > 3 * σ (i.e., perform 3σ-clipping).

Get updated $min[1], $max[1], $angle[1] quantities:

$min[1] = minimum($elon_work where $min[1] < $elon_work < $max[1])

$max[1] = maximum($elon_work where $min[1] < $elon_work < $max[1])

(where $min[1] and $max[1] in the r.h.s. of the above equations are the original values, before the update);

$angle[1] = $max[1] – $min[1]

III.1.iii. Sanity checks:

Should be true that, for each of index i = 0, 1: $angle[i] > 0;

abs($angle[i] - $recent) < $recent

($angle[i] and $recent should agree within 3.5 * $recent). Issue fatal error warning (die?) if violated.

III.1.iv. Final computations

To finalize computations, copy to new versions of the variables $min and $max:

$new_min = $min

$new_max = $max

If $new_min[0] < 0 then $new_min[0] = $new_min[0] + 360 deg

If $new_max[0] < 0 then $new_max[0] = $new_max[0] + 360 deg

If $new_min[1] < 0 then $new_min[1] = $new_min[1] + 360 deg

If $new_max[1] < 0 then $new_max[1] = $new_max[1] + 360 deg

$increment is a 2-element array of floating numbers; $Elon0 is as in the current survprogress script:

$increment[0] = $new_min[0] – $Elon0[1];

if $increment[0] < 0 then $increment[0] = $increment[0] + 360 deg;

subsequently, if $increment[0] > 180 deg then $increment[0] = $increment[0] – 360 degrees.

$increment[1] = $new_min[1] – $Elon0[0];

if $increment[1] < 0 then $increment[1] = $increment[1] + 360 deg;

subsequently, if $increment[1] > 180 deg then $increment[1] = $increment[1] – 360 degrees.

The computation of $FracS (a 2-element array for each of the two lunes) is:

for each of i = 0, 1, $FracS[i] = $angle[i] / 360.

Here the definition of the variables $new_min, $new_max (actually “$min,” “$max” in the existing survprogress script), and $angle = $max – $min are not necessarily the same as in the existing script.

For $increment[0] and $increment[1] both > 0 or both < 0, take the average of $increment[0] and $increment[1] to compute fractional increment:

**$Frac =** ($increment[0] + $increment[1]) / 360 deg + $FracS[0] + $FracS[1]

For $increment[0] > 0 and $increment[1] < 0, meaning that the lunes between $min[1] and $max[1] indicate an almost complete coverage:

$Frac = $increment[1] / 182.5 + $FracS[0] + $FracS[1]

For $increment[0] < 0 and $increment[1] > 0, meaning that the lunes between $min[0] and $max[0] indicate an almost complete coverage:

$Frac = $increment[0] / 182.5 + $FracS[0] + $FracS[1]

If **$Fra**c < 0.0 then **$Frac** = 1. + **$Frac**

$NCov = $epoch + int(**$Frac**)

$large_frac is a flag that, if set, indicates original **$Frac** was > 1.0 and thus coverage epoch

potentially > calendar epoch:

$large_frac = 0

If **$Frac** ≥ 1. then **$large_frac** = **$large_frac** + 1

If ($mjd1 – $mjd0) mod 182.5 (where the function x mod y is the remainder of x / y) is < 5 days (calendar epoch is just past anniversary) and **$NCov** – **$epoch** ≥ 1 then **$NCov** = **$NCov** – 1

(i.,e. coverage epoch needs to be decremented to agree with calendar one).

If **$Frac** > 1.0 then **$Frac** = **$Frac** – 1.0

The following is the case of calendar epoch just before anniversary but **$Frac** is small, indicating coverage epoch may need to be incremented:

If ($mjd1 - $mjd0) mod 182.5 is > 179 days (or current date is within 3.5 days of a semi-anniversary) and **$Frac** < 12% and **$large_frac** = 0, then **$NCov** = **$NCov** + 1.

The above two variables are the main output of new_survprogress.

Figs. 7 and 8 show an example of the effect on **$Frac** of 3σ-clipping and the removal of $elon when the rate of change of $elat as a function of time switches sign or when $elon exhibits a discontinuity, indicating crossing of one of the ecliptic poles. The estimate of **$Frac** is improved, vs. an overestimate when only 3σ-clipping is applied.

III.2. Notes on the algorithm of new_survprogress

Although Figs. 2—5 show the projection of lunes as colinear at their vertex, Fig.6 shows that the survey pattern does not maintain such co-linearity, per the dihedron angle of the NEOWISE-R mission. The algorithm does not rely on the collinearity of the lunes, except in the initial search for $min[1] and $max[1}. This search includes the addition of ±30 degrees padding, justified by the fact that the deviation from co-linearity is approximately 18.5 deg, per the NEOWISE Explanatory Supplement. The completion of sky coverage every approximately 6 months relies in actuality on when the initial ecliptic longitude of the mission is swept. In this way, inexactitudes of strictly semi-anniversaries are improved by the exact times when these sweeps occur.

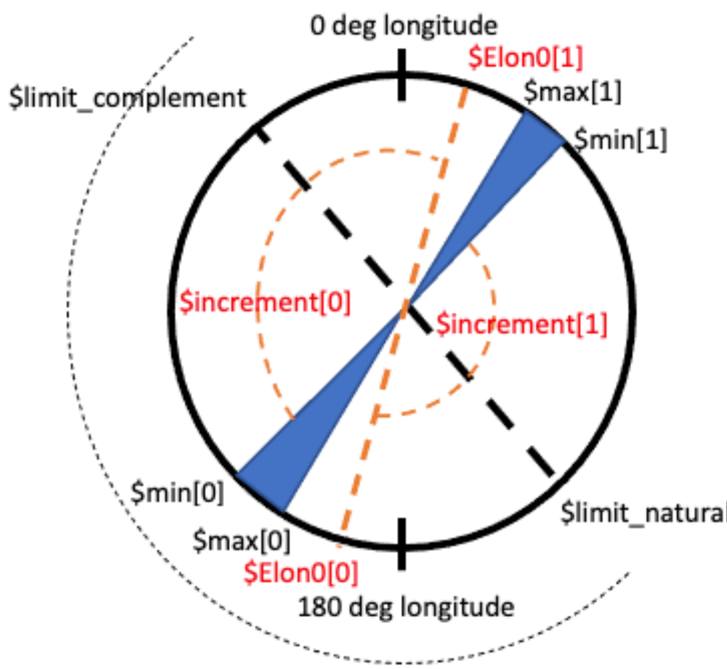

FIG. 2: Illustration of $limit_natural, as the mean Σ of ecliptic longitudes (shown as the blue lunes–actually seen in projection as sectors–swept by the scans within $scan_interval). In this case one set of longitudes is in the 2nd "quadrant" and the opposite set is in the 4th quadrant. All longitudes are in the range 0 – 360 degrees. The so-called $limit_complement is defined by $limit_natural – 180 degrees. The dotted curve is the range of longitudes where minima and maxima are searched for, to get $min[0] and $max[0]. The corresponding $min[1] and $max[1] are defined by adding 180 deg to each of $min[0] and $max[0]. Longitudes are checked for membership within each of the two lunes defined by $min and $max (that is, longitudes within each lune should have a span of value $angle within 0 of the value of $recent in degrees), and 3σ outliers are "clipped," in order to then iterate once. At the end of computations, $min and $max are converted to values in the range 0 – 360 deg, and the $increment angles (shown in red) are the difference between the beginning of each lune and the ecliptic longitudes of the lunes at the beginning of the mission (also shown in red, connected by a red straight dashed line, although they differ by about 185 deg). All angles are positive in the counterclockwise direction.

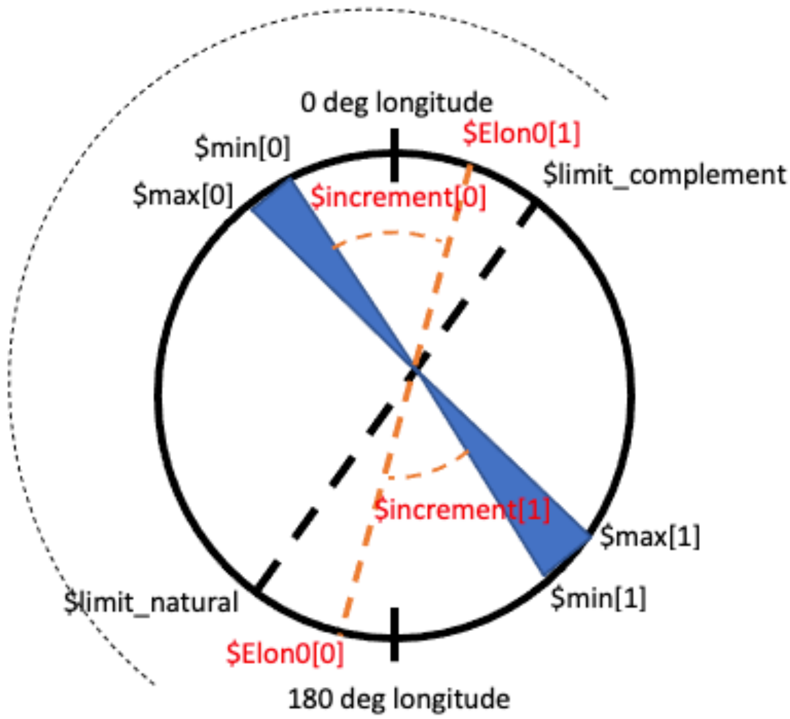

FIG. 3: Illustration of $limit_natural, as the mean Σ of ecliptic longitudes (shown as the blue lunes–actually seen in projection as sectors–swept by the scans within $scan_interval). In this case one set of longitudes is in the 1st "quadrant" and the opposite set is in the 3[th] quadrant. All NEOWISE observed longitudes are in the range 0 – 360 degrees. The so-called $limit_complement is defined by $limit_natural – 180 deg and is < 0 in this case. The dotted curve is the range of longitudes where minima and maxima are searched for, to get $min[0] and $max[0]. This range of longitudes includes negative ones in the interval $limit_complement to 0 deg longitude, so that the whole range is continuous (at or near 0 deg) in this interval. The corresponding $min[1] and $max[1] are defined by adding 180 deg to each of $min[0] and $max[0]. Longitudes are checked for membership within each of the two lunes defined by $min and $max (that is, longitudes within each lune should have a span of value $angle within 30% of the value of $recent in degrees), and 3σ outliers are "clipped," in order to then iterate once. At the end of computations, $min and $max are converted to values in the range 0 – 360 deg, and the $increment angles (shown in red) are the difference between the beginning of each lune and the ecliptic longitudes of lunes at the beginning of the mission (also shown in red, connected by a red straight dashed line, although they differ by about 185 deg). All angles are positive in the counterclockwise direction.

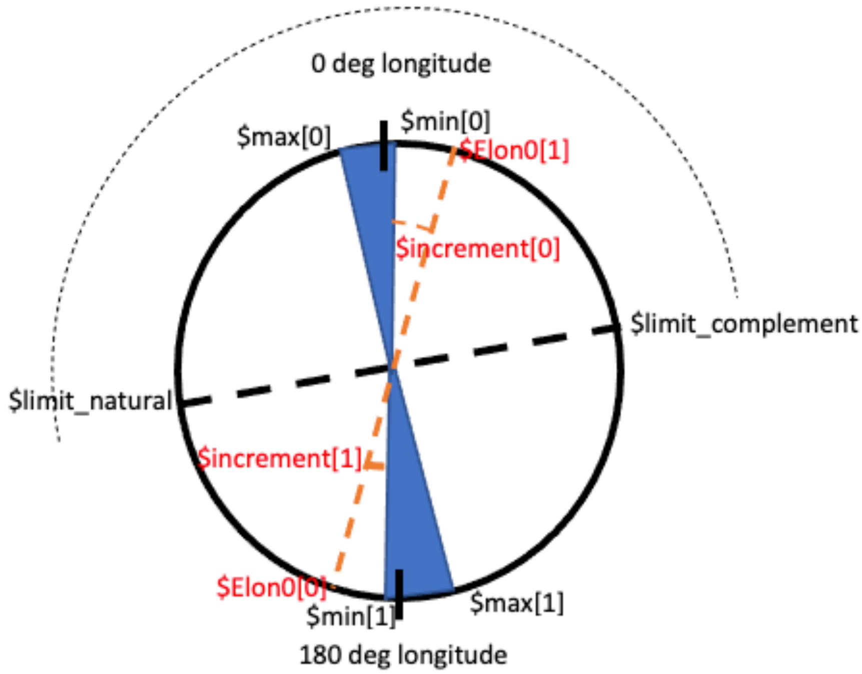

FIG. 4: Illustration of $limit_natural, as the mean Σ of ecliptic longitudes (shown as the blue lunes–actually seen in projection as sectors–swept by the scans within $scan_interval). In this case one set of longitudes is mostly in the 1st "quadrant" but also somewhat in the 4th quadrant, straddling 0 deg longitude. The opposite set is mostly in the 3rd quadrant but also somewhat in the 2nd quadrant, straddling 180 deg longitude. Most NEOWISE observed longitudes are in the range 0 – 360 deg, except for those between $min[0] and 0 deg longitude, from which 360 deg was subtracted to make them negative. The so-called $limit_complement is defined by $limit_natural – 180 deg and is < 0 in this case. The dotted curve is the range of longitudes where minima and maxima are searched for, to get $min[0] and $max[0]. This range of longitudes includes negative ones in the interval $limit_complement to 0 deg longitude, so that the whole range is continuous (at or near 0 deg) in this interval. The corresponding $min[1] and $max[1] are defined by adding 180 deg to each of $min[0] and $max[0]. Longitudes are checked for membership within each of the two lunes defined by $min and $max (that is, longitudes within each lune should have a span of value $angle within 30% of the value of $recent in degrees), and 3σ outliers are "clipped," in order to then iterate once. At the end of computations, $min and $max are converted to values in the range 0 – 360 deg, and the $increment angles (very small angles in this case, shown in red) are the difference between the beginning of each lune and the ecliptic longitudes of the lunes at the beginning of the mission (also shown in red, connected by a red straight dashed line, although they differ by about 185 deg). All angles are positive in the counterclockwise direction.

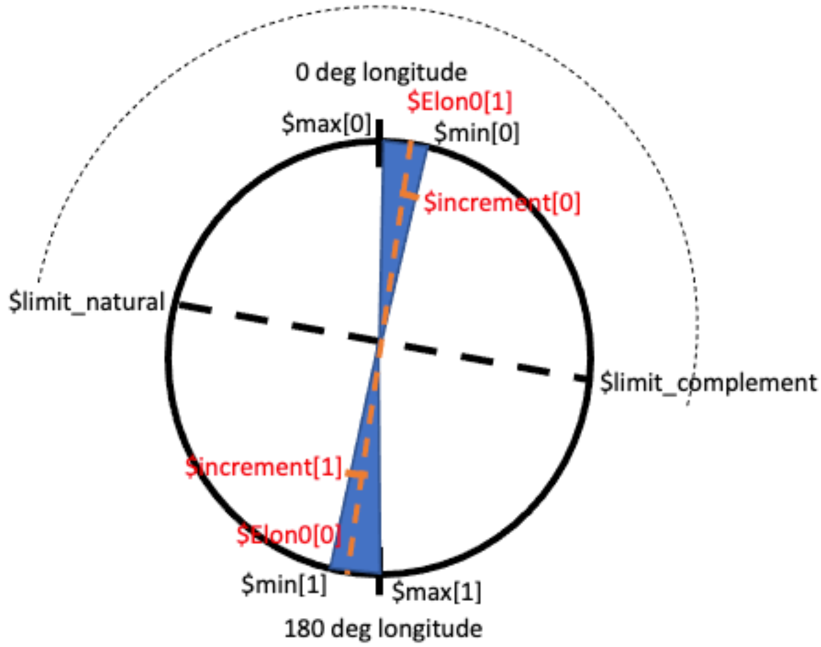

FIG. 5: Illustration of $limit_natural, as the mean Σ of ecliptic longitudes (shown as the blue lunes–actually seen in projection as sectors–swept by the scans within $scan_interval). In this case one set of longitudes is in the 2nd "quadrant" and the opposite set is in the 4th quadrant. However, unlike Fig. 4, NEOWISE observed longitudes close to 360 deg approach, but never reach 360 degrees. Similarly, NEOWISE observed longitudes close to 180 deg do not exceed the latter. Longitudes within 2*$recent of 360 deg but shortward of the latter have 360 deg subtracted to make them negative. The so-called $limit_complement is defined by $limit_natural – 180 deg and is < 0 in this case. The dotted curve is the range of longitudes where minima and maxima are searched for, to get $min[0] and $max[0]. This range of longitudes includes negative ones in the interval $limit_complement to 0 deg longitude. The corresponding $min[1] and $max[1] are defined by adding 180 deg to each of $min[0] and $max[0]. Longitudes are checked for membership within each of the two lunes defined by $min and $max (that is, longitudes within each lune should have a span of value $angle within 30% of the value of $recent in degrees), and 3σ outliers are "clipped," in order to then iterate once. At the end of computations, $min and $max are converted to values in the range 0 – 360 deg, and the $increment angles (very small angles in this case, shown in red) are the difference between the beginning of each lune and the ecliptic longitudes of the lunes at the beginning of the mission (also shown in red, connected by a red straight dashed line, although they differ by about 185 deg). All angles are positive in the counterclockwise direction; the $increment small angles in this case are negative.

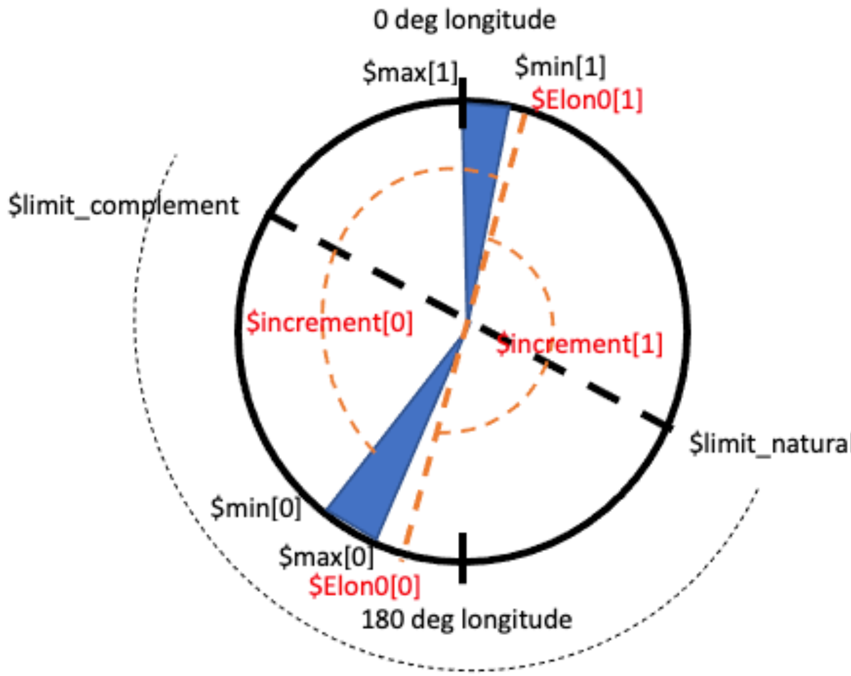

FIG. 6: Illustration of the fact that the two lunes (shown in projection as sectors, swept by the scans within $scan_interval) are not necessarily co-linear at their common vertex. Figs. 2—5 show the two sectors as co-linear, as if their vertex were exactly at the ecliptic pole, which realistically is not the case due to the dihedron angle of the NEOWISE-R mission. See Figs. 2—5 for a detailed explanation of the features and legends of this type of depiction.

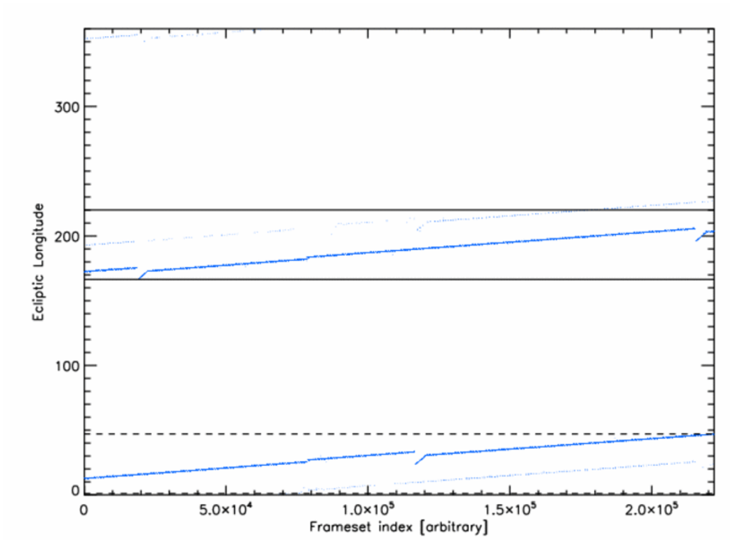

Fig. 7: Ecliptic longitudes as a function of arbitrary frameset index (equivalent to time), for an example trending interval of 34 days ending on UT 2023 Jan 16. The blue symbols are $elon from getfix for 221,966 framesets. The two horizontal dashed lines (the bottom one very close to the x-axis) are $min[0] and $max[0], and demarcate a corresponding lune as in Figs. 2—6. The horizontal solid lines are $min[1] and $max[1] and also demarcate the lune in the opposite side of the sky. All $elon values are plotted in the same shade of blue, but of order 1 frameset per scan has discontinuous $elon values when an ecliptic pole is crossed, and appear as "light blue" symbols below or above the locus of most $elon values (which appear as dark blue lines). Some outliers were removed by 3σ-clipping, and the resulting fractional coverage for Year 9 was measured at 26%. Fig. 8 demonstrates a truer measurement of this value.

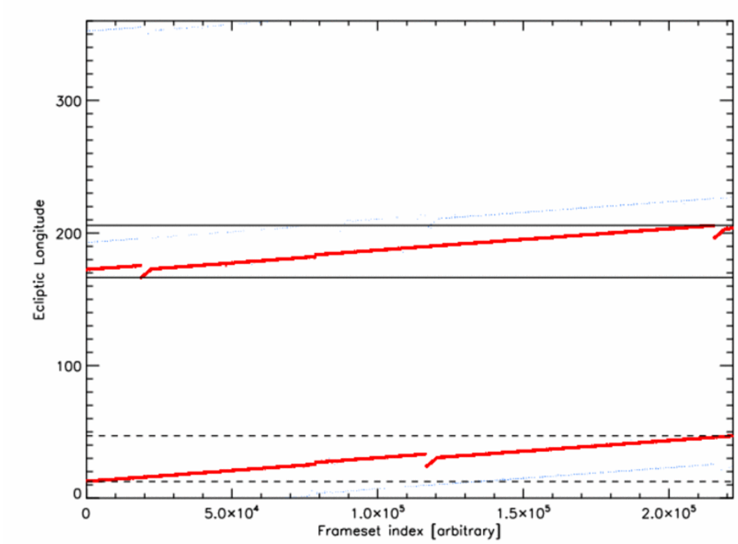

Fig. 8: Same as Fig. 5, except that superposed as red symbols are $elon_work values where outliers have been removed not only by 3σ-clipping, but also when ecliptic latitude switches rate of change (as a function of time), and when $elon jumps by > 2 degrees between adjacent framesets, within the 15-percentile and 85-percentile of each scan. The horizontal dashed lines are $min[0] and $max[0], and the horizontal solid lines are $min[1] and $max[1], as determined from $elon_work. Most values of $elon coincide with $elon_work, but note how the "light blue" values of $elon are discarded in the $elon_work sample. The resulting fractional coverage for Year 9 is measured at 19% and is a truer representation than in Fig. 7.

Version 1.0 - Last updated: 8 March 2023

S. Fajardo-Acosta & David Imel - IPAC