

Pygac 2-step slice

Source code translation: $s_t = \text{temp_start_line}$
 $e_t = \text{temp_end_line}$
 $s_u = \text{start_line}$
 $e_u = \text{end_line}$

General procedure:

- Choose temporary start/end lines s_t/e_t due to invalid lat/lon info:

$$\text{data} = \text{data}[s_t : e_t]$$

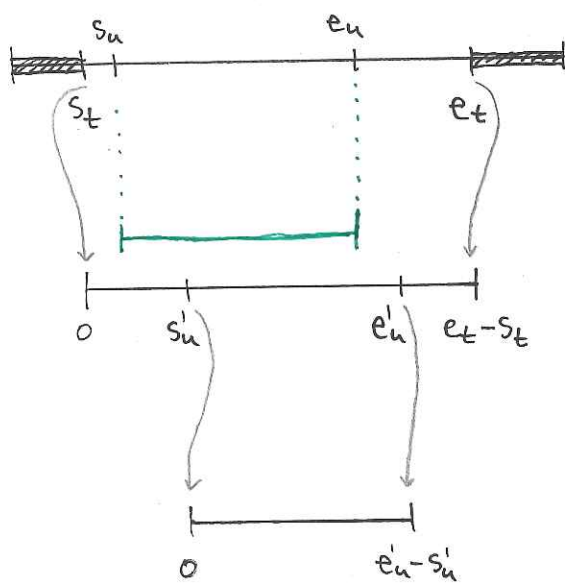
- Update scanline range s_u, e_u requested by the user to the new slice:

$$s_u \rightarrow s'_u, e_u \rightarrow e'_u \quad \parallel \text{Erroneous}$$

- Select user scanline range

$$\text{data} = \text{data}[s'_u : e'_u]$$

The bug only occurs, if $e_t > e_u$ and $e_u < e_t - s_t$:



Orig. Data ||||| = invalid lat/lon info

↓
slice 1 ($s_t : e_t$)

↓
slice 2 ($s'_u = s_u : e'_u = e_u$): Here is the bug. The values of s_u and e_u are unchanged, but now they are applied to a smaller array. The green interval indicates the correct slice. As you can see, the final slice is shifted to the right.

If $e_t > e_u$, but $e_u > e_t - s_t$, pygac throws an `IndexError` (access to index beyond array bounds)

Consequences for CDARA-A2:

- S_u is always 0 \rightarrow no problem
- $E_u <$ along track to avoid overlap \rightarrow Problem:

