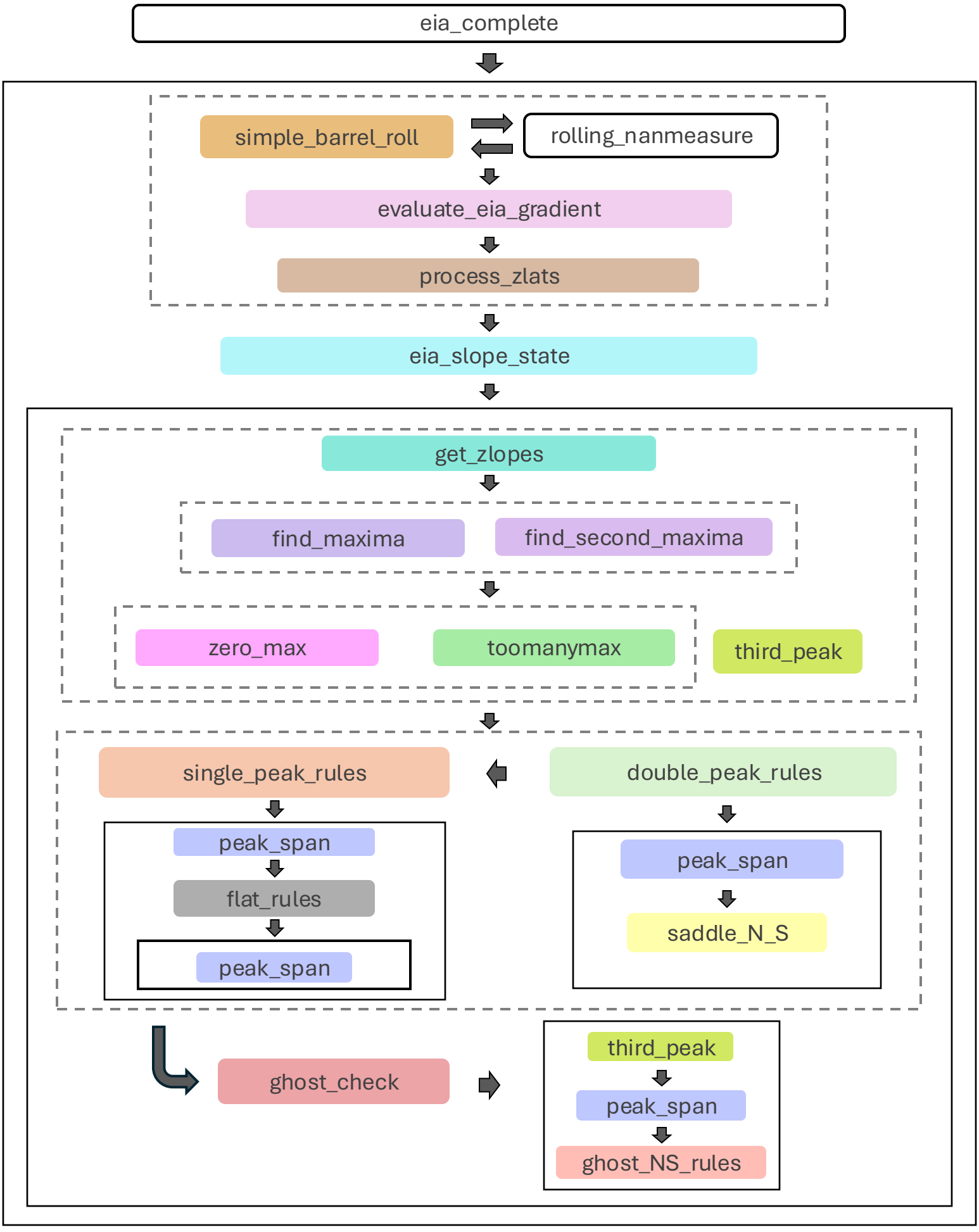
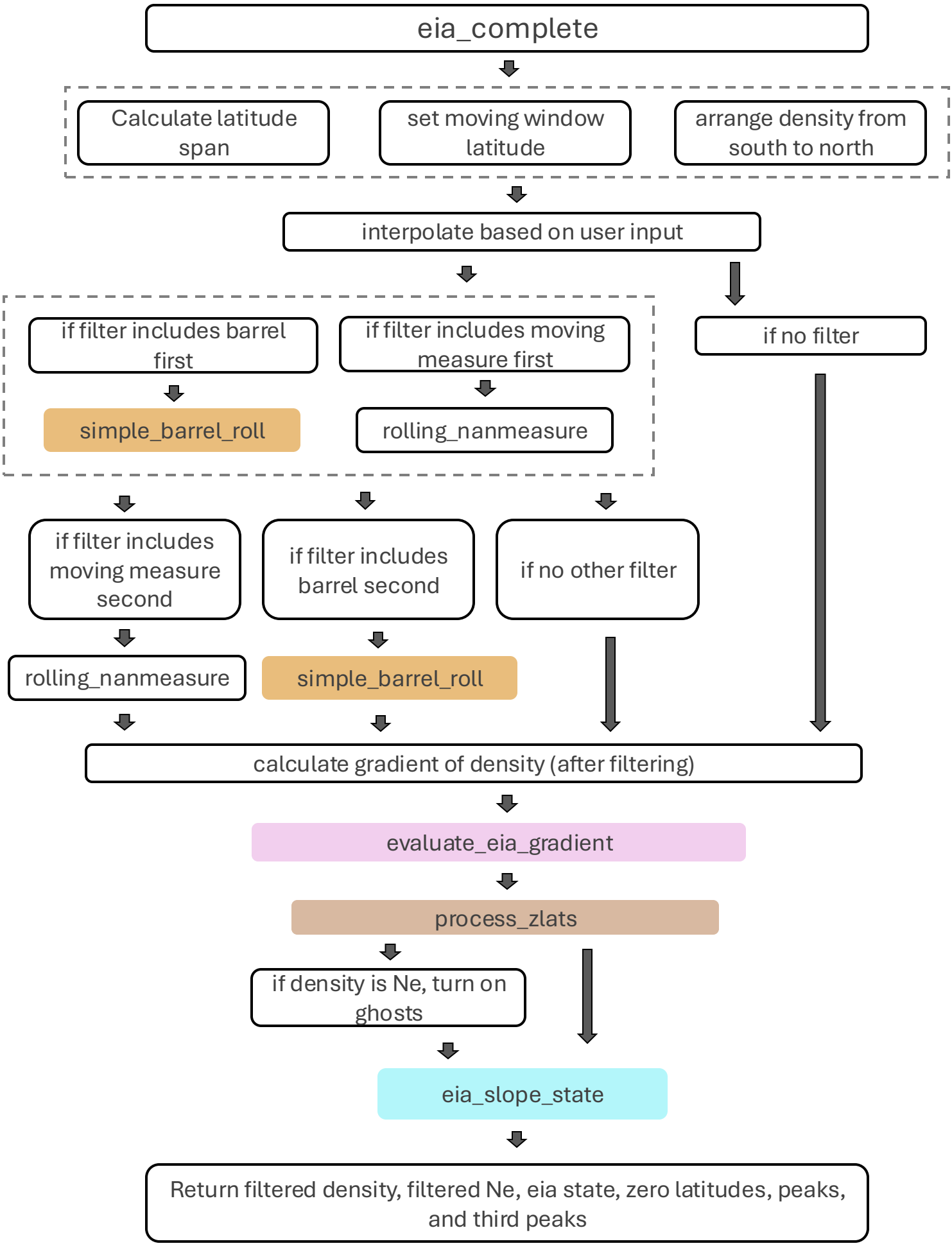
EIA Detection Flow Charts


eia_complete
simple_barrel_roll
rolling_nanmeasure
evaluate_eia_gradient
process_zlats
eia_slope_state
get_zlopes
find_maxima
find_second_maxima
zero_max
toomanymax
third_peak
single_peak_rules
double_peak_rules
peak_span
flat_rules
peak_span
peak_span
saddle_N_S
ghost_check
third_peak
peak_span
ghost_NS_rules
KEY
Related
Code Name
Grouped in same code


eia_complete
Calculate latitude span
set moving window latitude
arrange density from south to north
interpolate based on user input
if filter includes barrel first
if filter includes moving measure first
if no filter
simple_barrel_roll
rolling_nanmeasure
if filter includes moving measure second
if filter includes barrel second
if no other filter
rolling_nanmeasure
simple_barrel_roll
calculate gradient of density (after filtering)
evaluate_eia_gradient
process_zlats
if density is Ne, turn on ghosts
eia_slope_state
Return filtered density, filtered Ne, eia state, zero latitudes, peaks, and third peaks

Simple barrel roll

Set first points as starting x and y contact points

While the index is less than the last point in the array, look for contact points

Set the barrel radius from input

Establish a region of interest within barrel radius

Calculate the angular distance between current contact and region of interest points

Choose smallest angular distance as next contact point

Linearly interpolate contacts to be the same length as original array

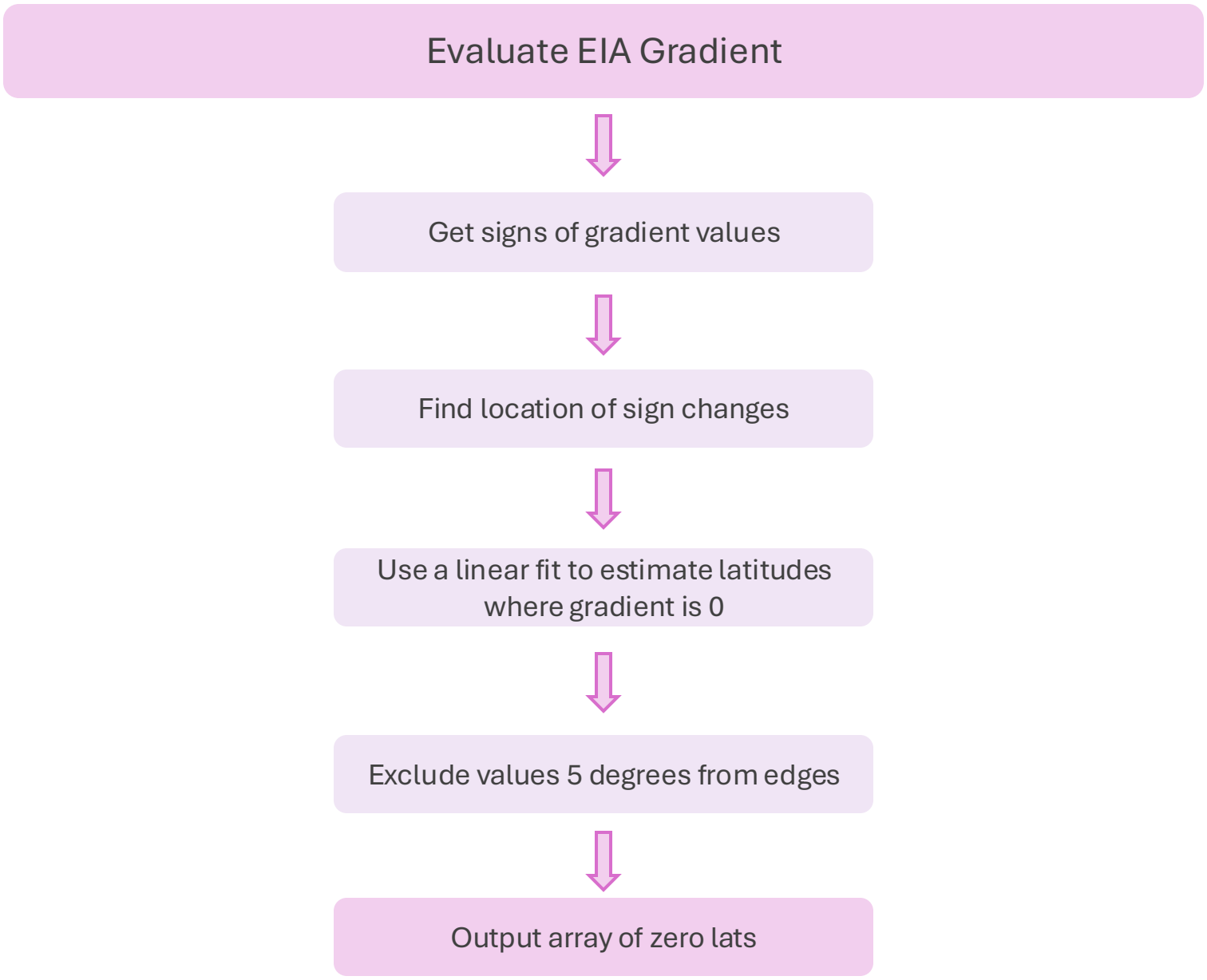
If envelope = 1

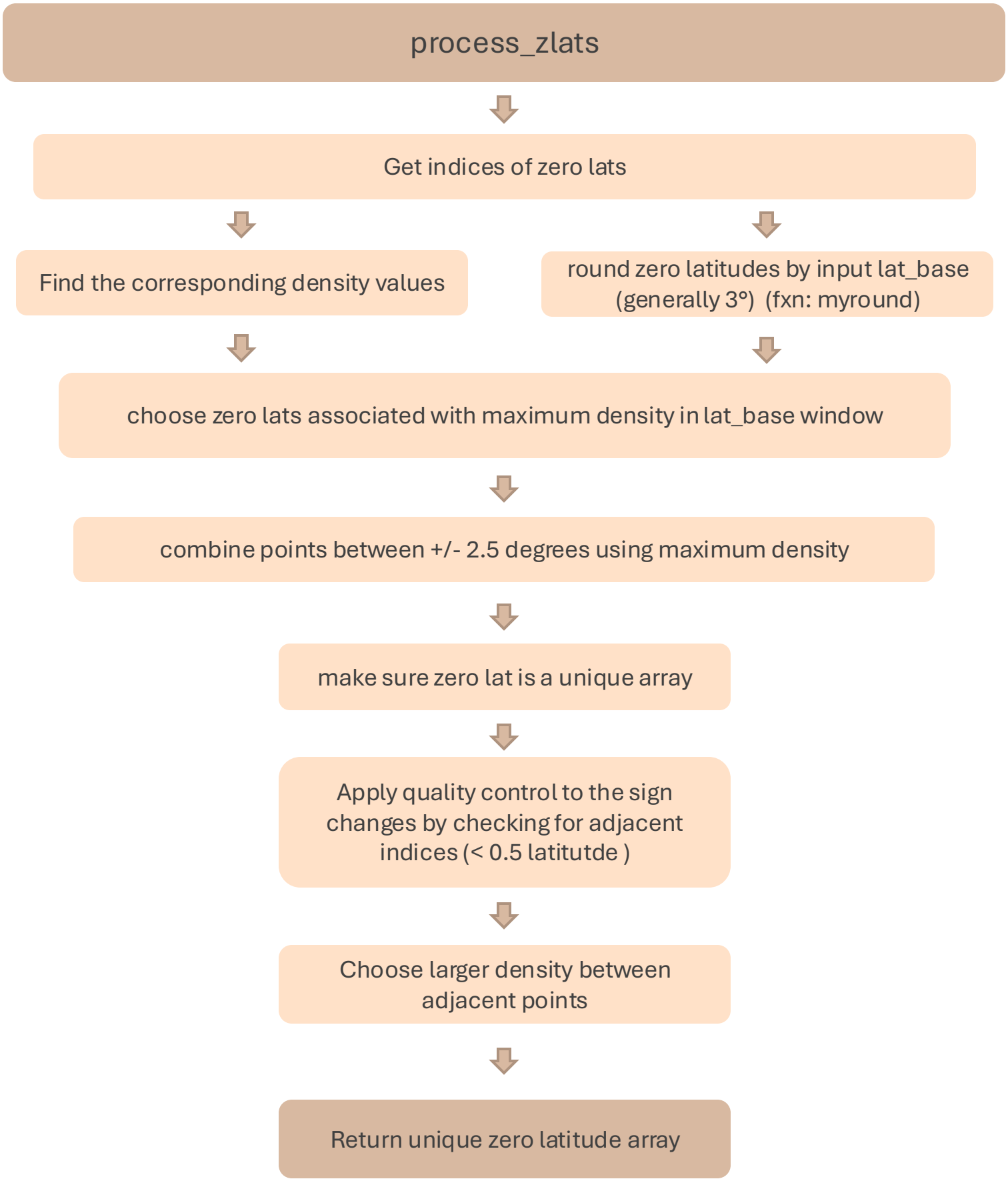
If envelope != 1

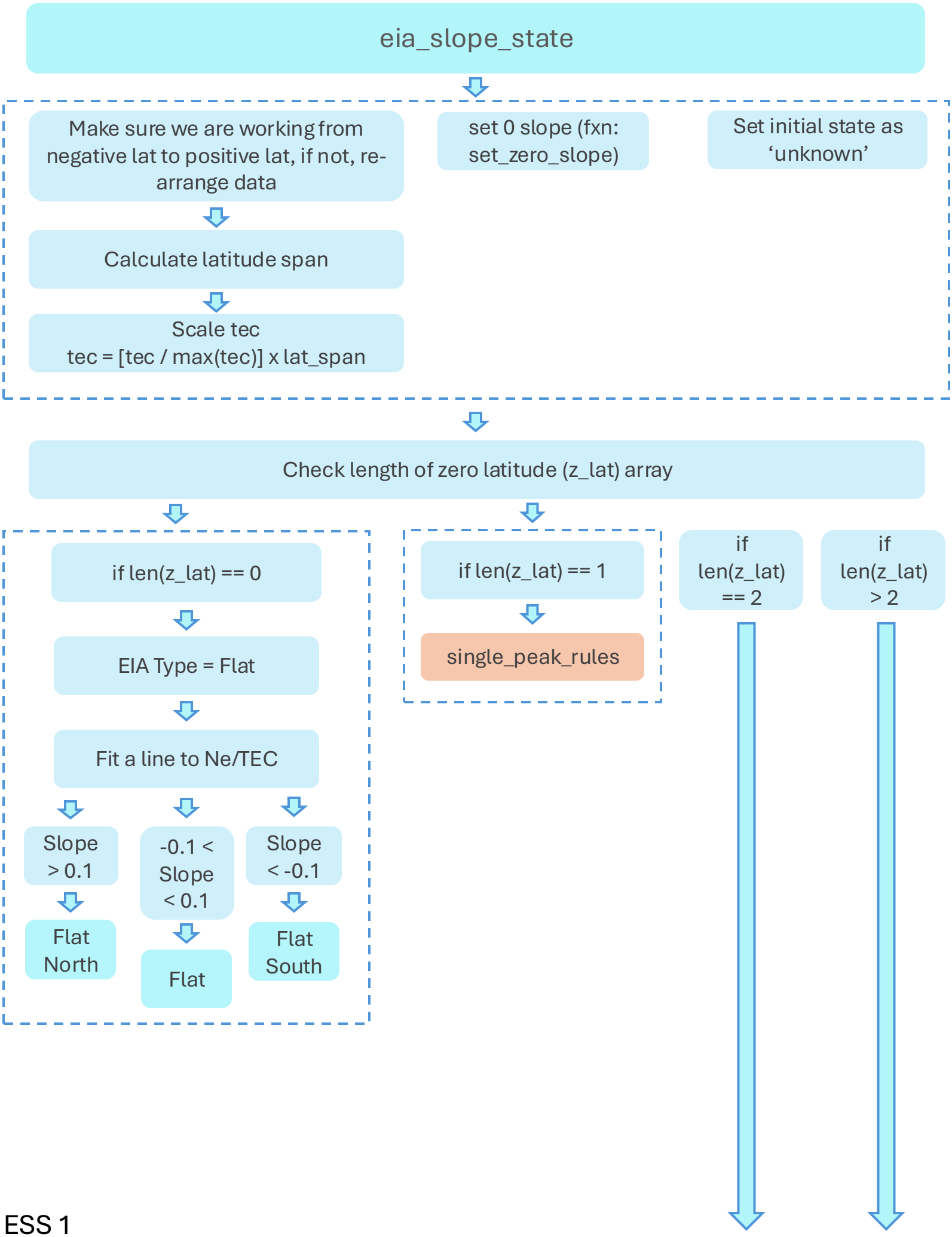
Include points between +0.02*max y and – 0.06* min y, rescale array

Do not use an envelope, return array that is rescaled

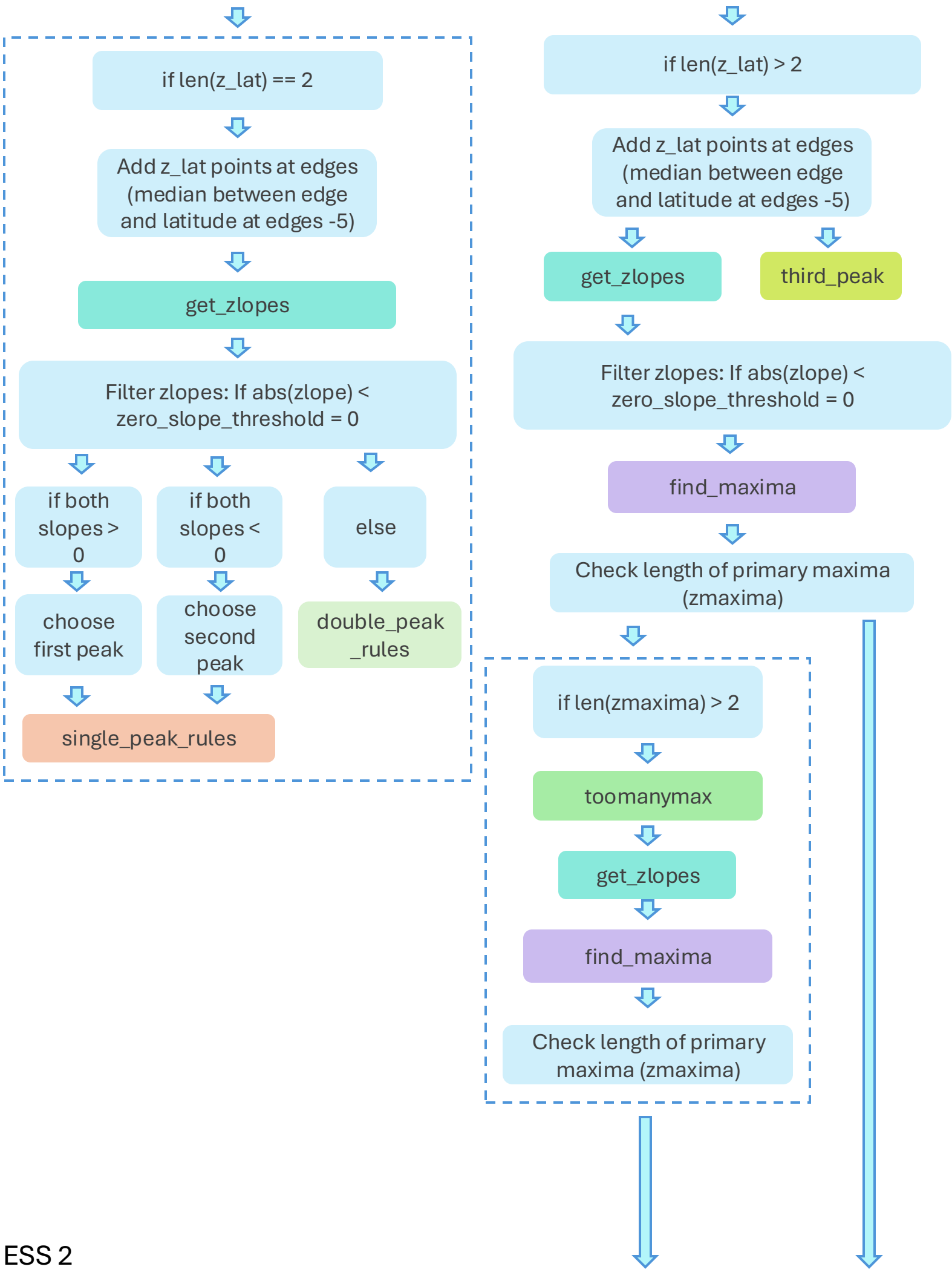
Output barrel-rolled data


Evaluate EIA Gradient
Get signs of gradient values
Find location of sign changes
Use a linear fit to estimate latitudes where gradient is 0
Exclude values 5 degrees from edges
Output array of zero lats

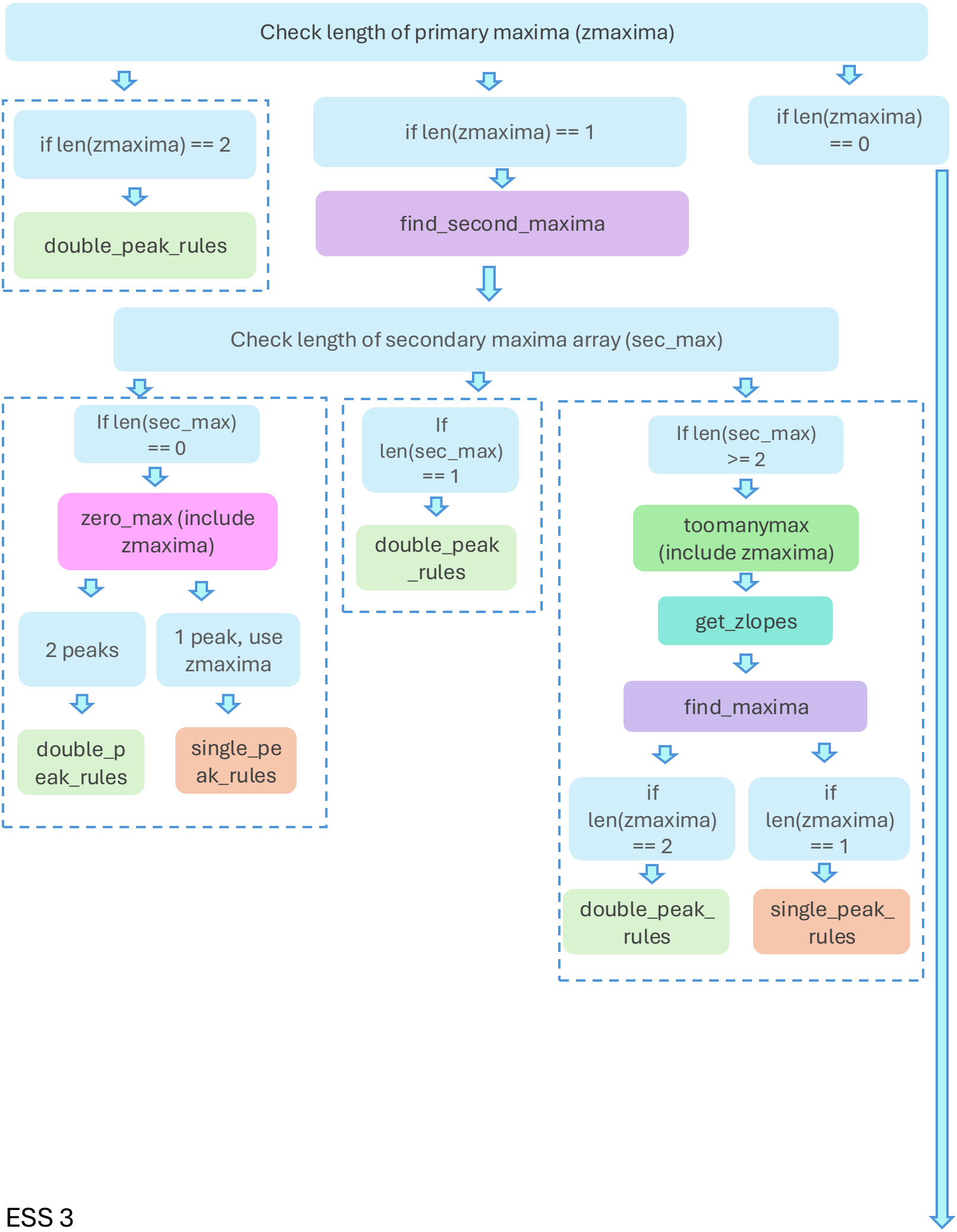

process_zlats
Get indices of zero lats
Find the corresponding density values
round zero latitudes by input lat_base (generally 3°) (fxn: myround)
choose zero lats associated with maximum density in lat_base window
combine points between +/- 2.5 degrees using maximum density
make sure zero lat is a unique array
Apply quality control to the sign changes by checking for adjacent indices (< 0.5 latitutde)
Choose larger density between adjacent points
Return unique zero latitude array


eia_slope_state
Make sure we are working from negative lat to positive lat, if not, re-arrange data
Calculate latitude span
Scale tec
tec = [tec / max(tec)] x lat_span
set 0 slope (fxn: set_zero_slope)
Set initial state as 'unknown'
Check length of zero latitude (z_lat) array
if len(z_lat) == 0
EIA Type = Flat
Fit a line to Ne/TEC
Slope > 0.1
Flat North
-0.1 < Slope < 0.1
Flat
Slope < -0.1
Flat South
if len(z_lat) == 1
single_peak_rules
if len(z_lat) == 2
if len(z_lat) > 2

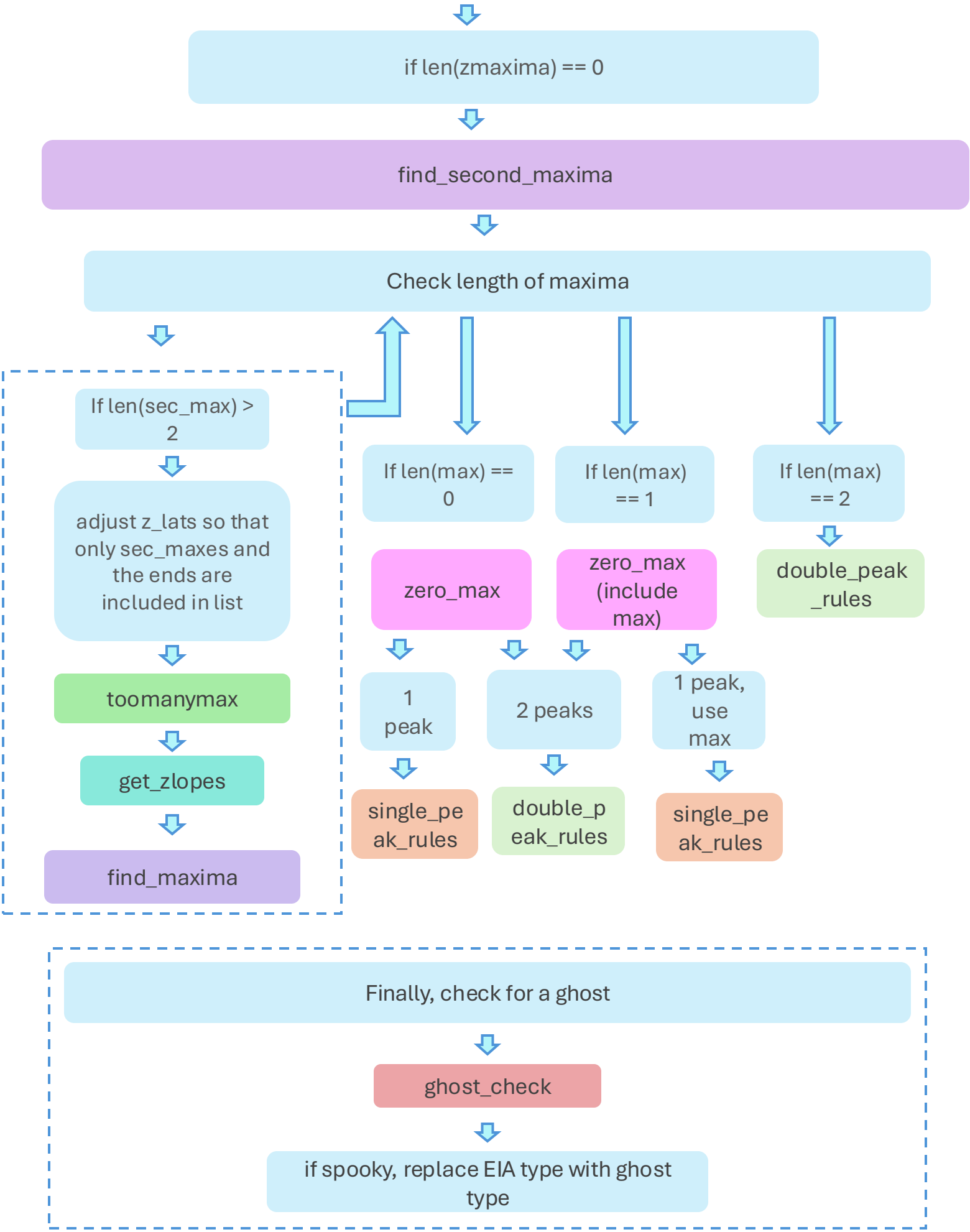
ESS 1


if len(z_lat) == 2
Add z_lat points at edges (median between edge and latitude at edges -5)
get_zlopes
Filter zlopes: If abs(zlope) < zero_slope_threshold = 0
if both slopes > 0
if both slopes < 0
else
choose first peak
choose second peak
double_peak _rules
single_peak_rules
if len(z_lat) > 2
Add z_lat points at edges (median between edge and latitude at edges -5)
get_zlopes
third_peak
Filter zlopes: If abs(zlope) < zero_slope_threshold = 0
find_maxima
Check length of primary maxima (zmaxima)
if len(zmaxima) > 2
toomanymax
get_zlopes
find_maxima
Check length of primary maxima (zmaxima)

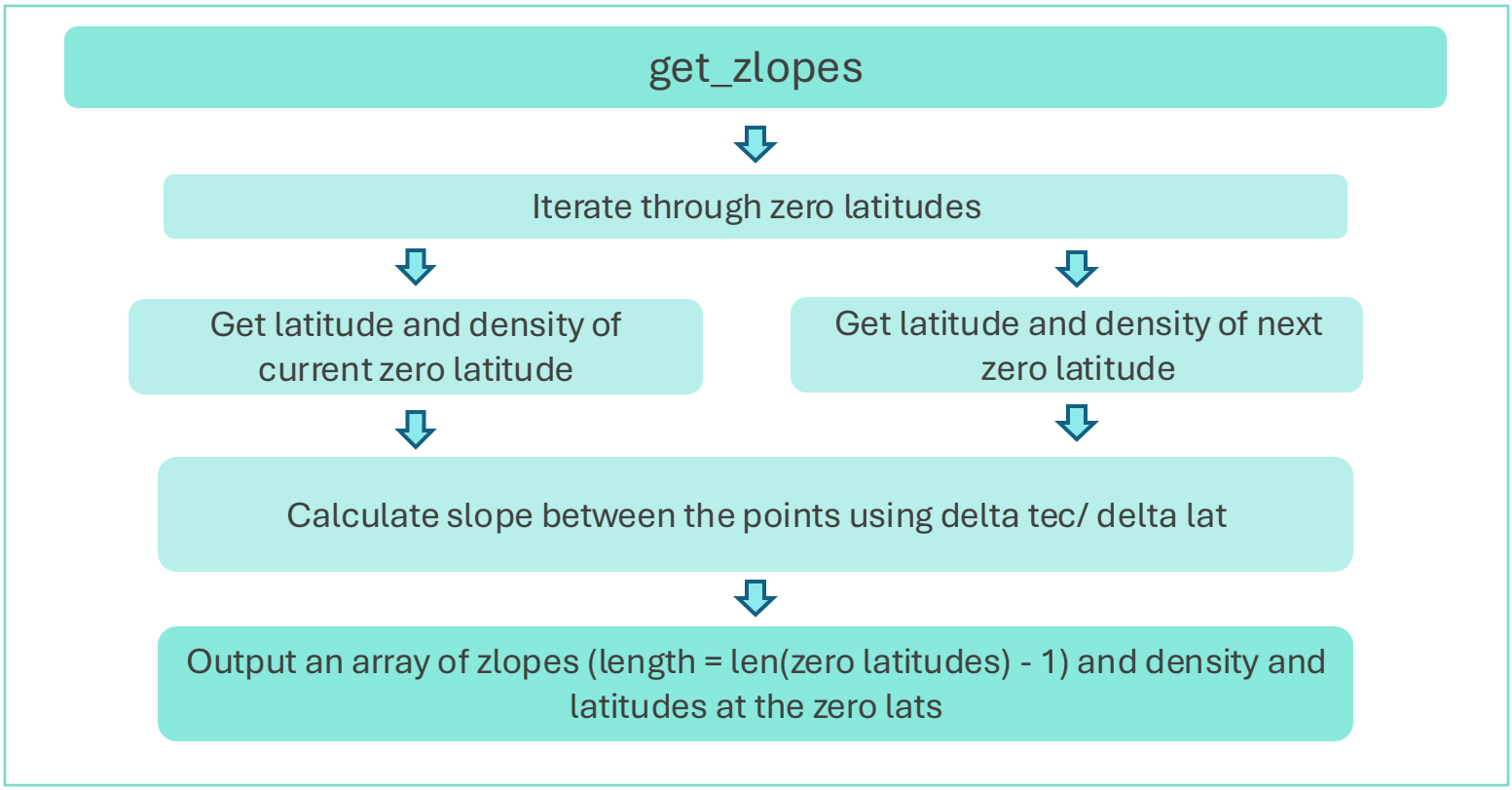
ESS 2

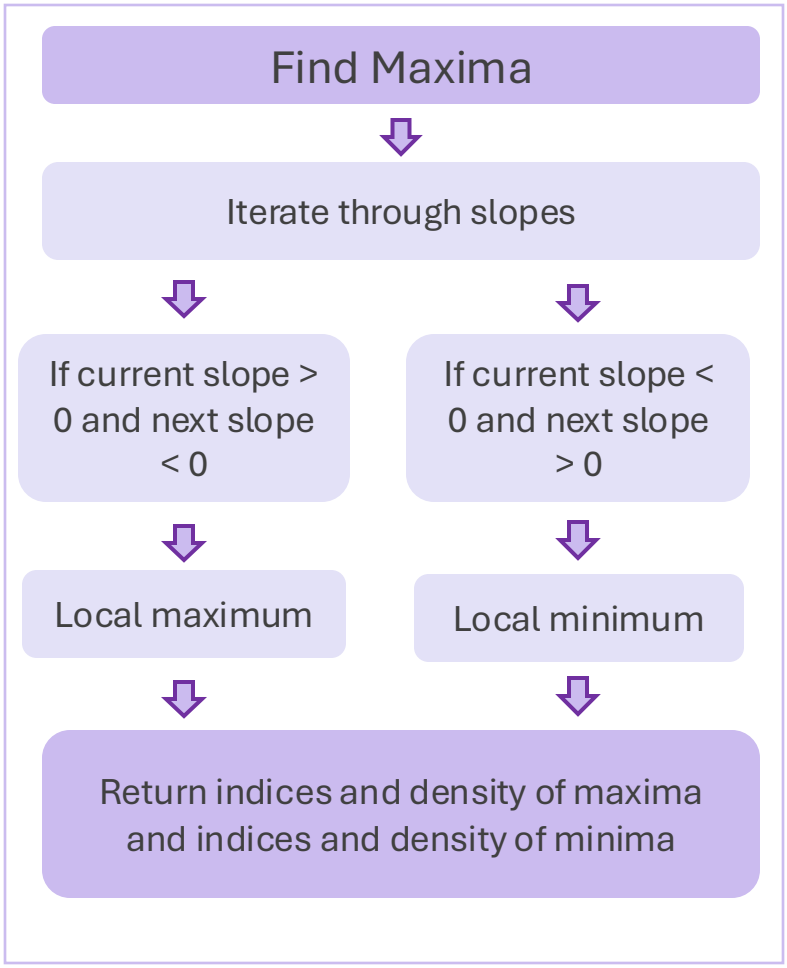

Check length of primary maxima (zmaxima)
if len(zmaxima) == 2
double_peak_rules
if len(zmaxima) == 1
find_second_maxima
if len(zmaxima) == 0
Check length of secondary maxima array (sec_max)
If len(sec_max) == 0
zero_max (include zmaxima)
2 peaks
1 peak, use zmaxima
double_peak_rules
single_peak_rules
If len(sec_max) == 1
double_peak_rules
If len(sec_max) >= 2
toomanymax (include zmaxima)
get_zlopes
find_maxima
if len(zmaxima) == 2
if len(zmaxima) == 1
double_peak_rules
single_peak_rules

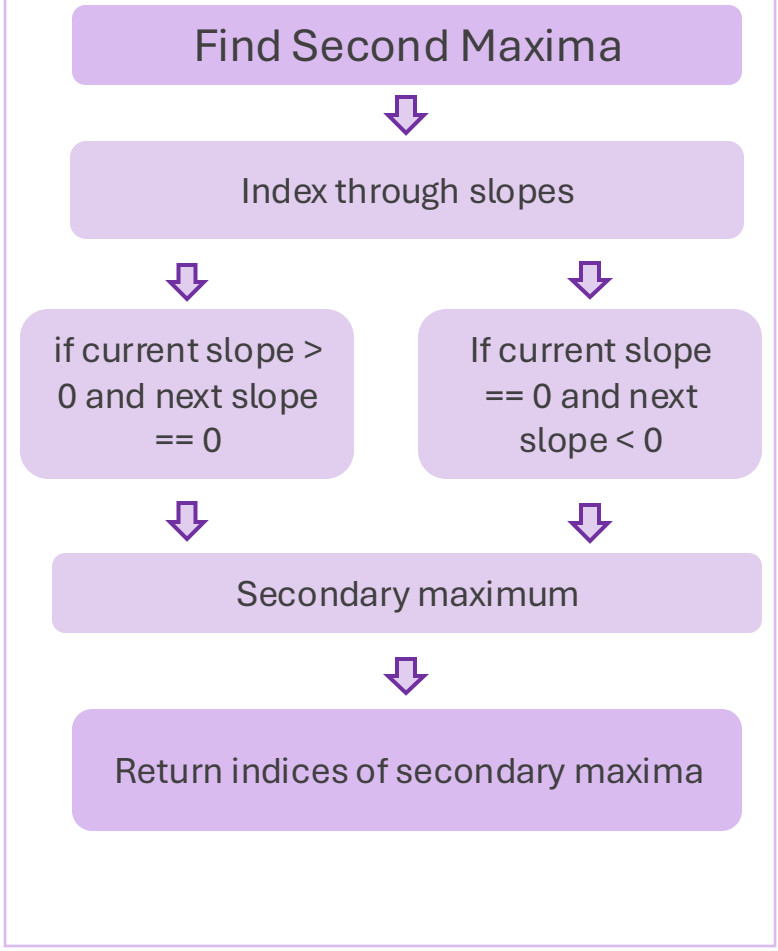
ESS 3

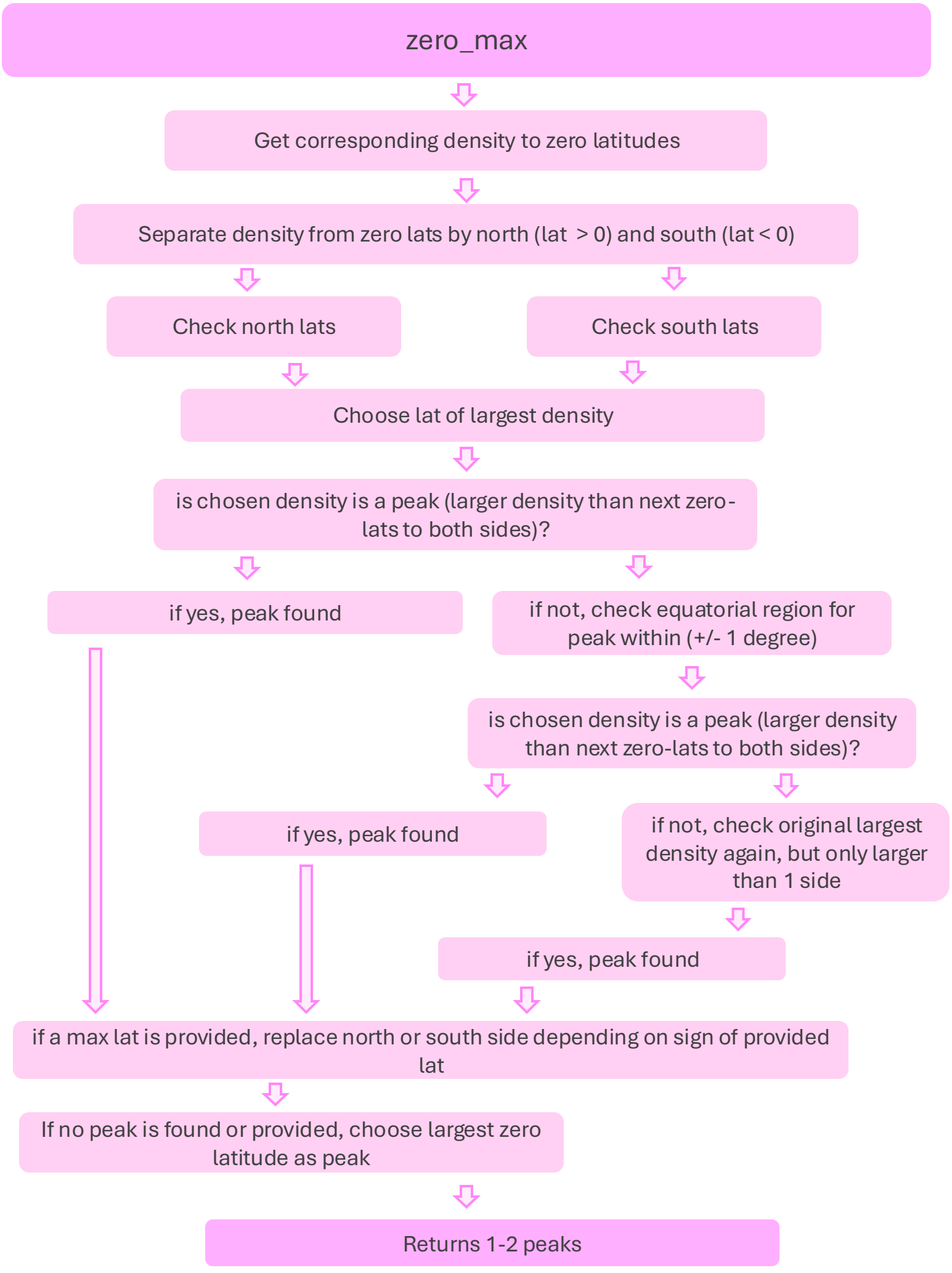

if len(zmaxima) == 0
find_second_maxima
Check length of maxima
If len(sec_max) > 2
adjust z_lats so that only sec_maxes and the ends are included in list
toomanymax
get_zlopes
find_maxima
If len(max) == 0
If len(max) == 1
If len(max) == 2
zero_max
zero_max (include max)
double_peak_rules
1 peak
2 peaks
1 peak, use max
single_peak_rules
double_peak_rules
single_peak_rules
Finally, check for a ghost
ghost_check
if spooky, replace EIA type with ghost type

ESS 4


get_zlopes
Iterate through zero latitudes
Get latitude and density of current zero latitude
Get latitude and density of next zero latitude
Calculate slope between the points using delta tec/ delta lat
Output an array of zlopes (length = len(zero latitudes) - 1) and density and latitudes at the zero lats


Find Maxima
Iterate through slopes
If current slope > 0 and next slope < 0
If current slope < 0 and next slope > 0
Local maximum
Local minimum
Return indices and density of maxima and indices and density of minima


Find Second Maxima
Index through slopes
if current slope > 0 and next slope == 0
If current slope == 0 and next slope < 0
Secondary maximum
Return indices of secondary maxima


zero_max
Get corresponding density to zero latitudes
Separate density from zero lats by north (lat > 0) and south (lat < 0)
Check north lats
Check south lats
Choose lat of largest density
is chosen density is a peak (larger density than next zero-lats to both sides)?
if yes, peak found
if not, check equatorial region for peak within (+/- 1 degree)
is chosen density is a peak (larger density than next zero-lats to both sides)?
if yes, peak found
if not, check original largest density again, but only larger than 1 side
if yes, peak found
if a max lat is provided, replace north or south side depending on sign of provided lat
If no peak is found or provided, choose largest zero latitude as peak
Returns 1-2 peaks

toomanymax

find corresponding density to zero latitudes

Separate tec from zero lats by north (between 3 and 20 mlat), south (between -3 and -20 mlat), and equator (between between +/-3)

if there are north lats

If no other lat provided or provided lat is south

Choose lat closest to equator in north lats

If north lat provided

Use provided lat

If there are equator lats

Choose lat of largest density

If there are south lats

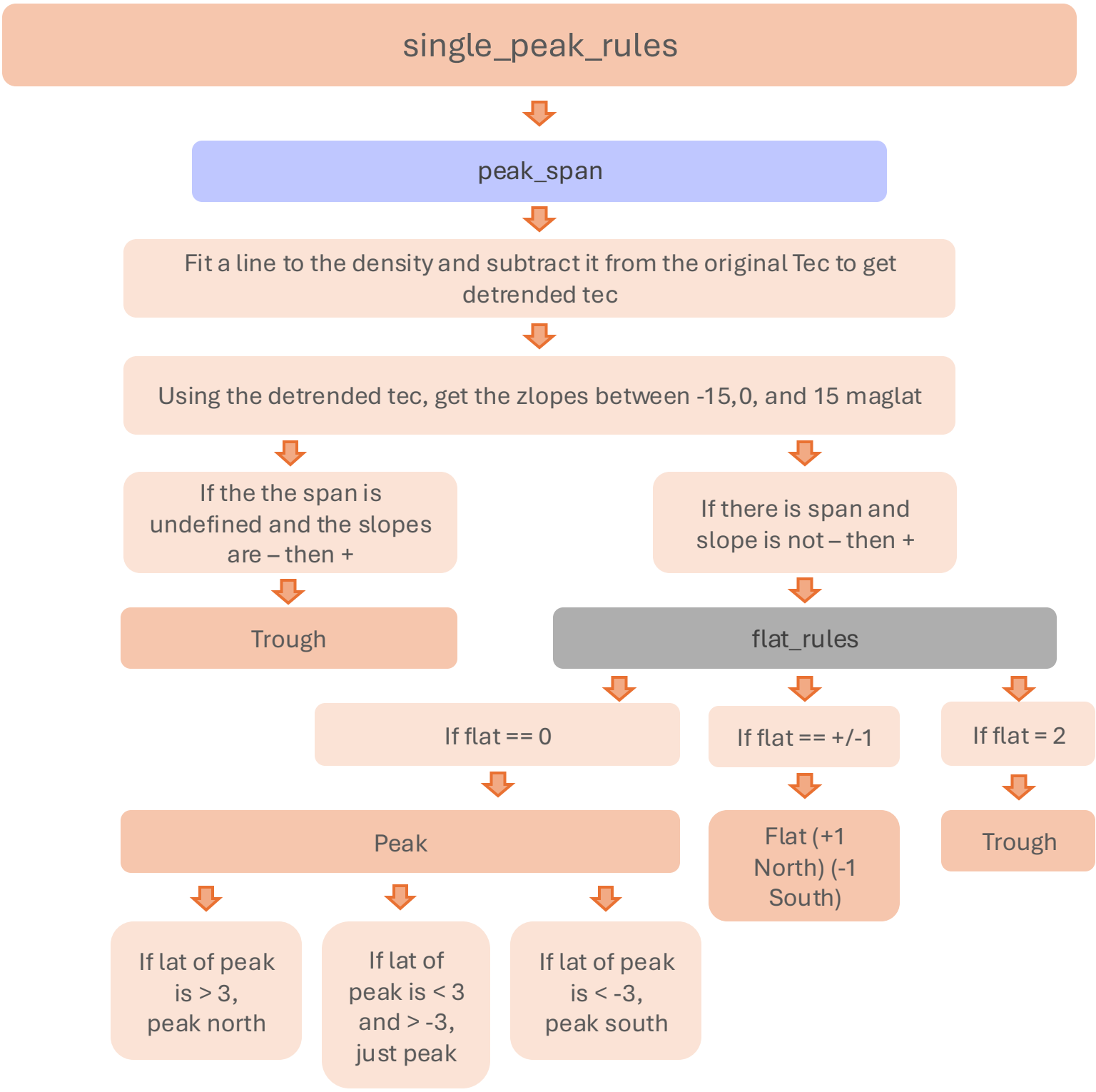
If no other lat provided or provided lat is north

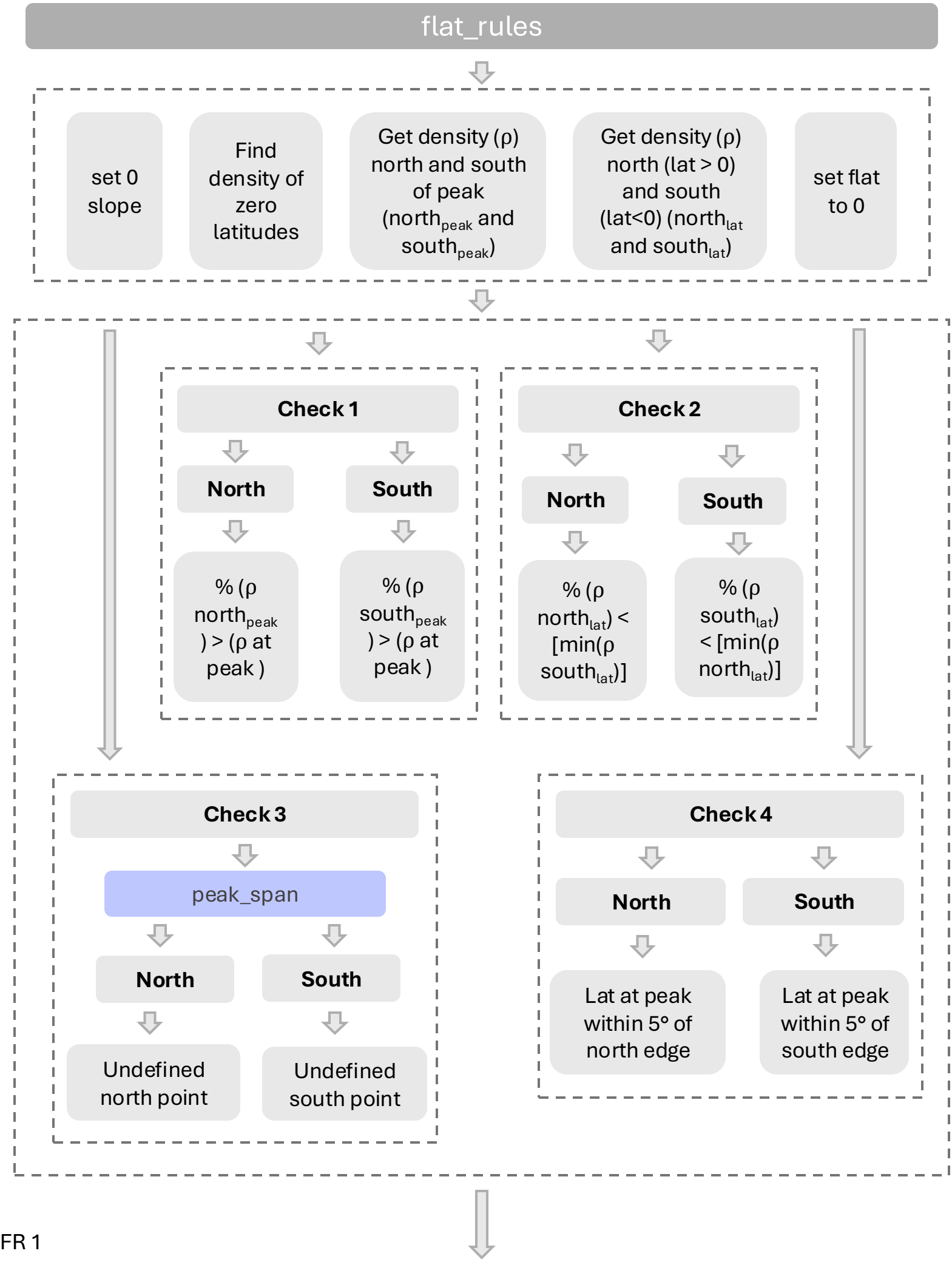
Choose lat closest to equator in south lats

If north lat provided

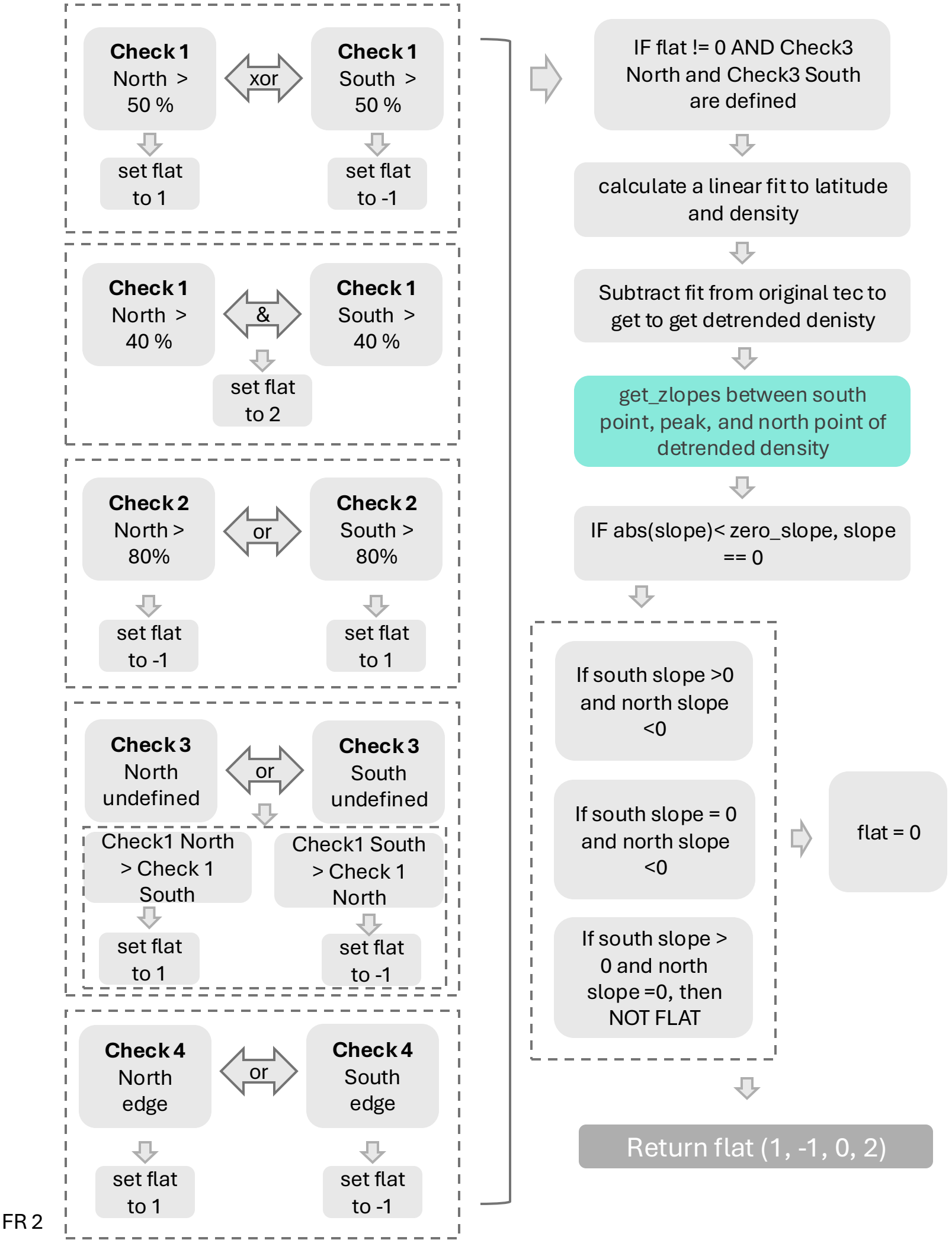
Use provided lat

Return unique new latitude array that contains a maximum of 5 values [south edge, closest south, equator max, closest north, and north edge]

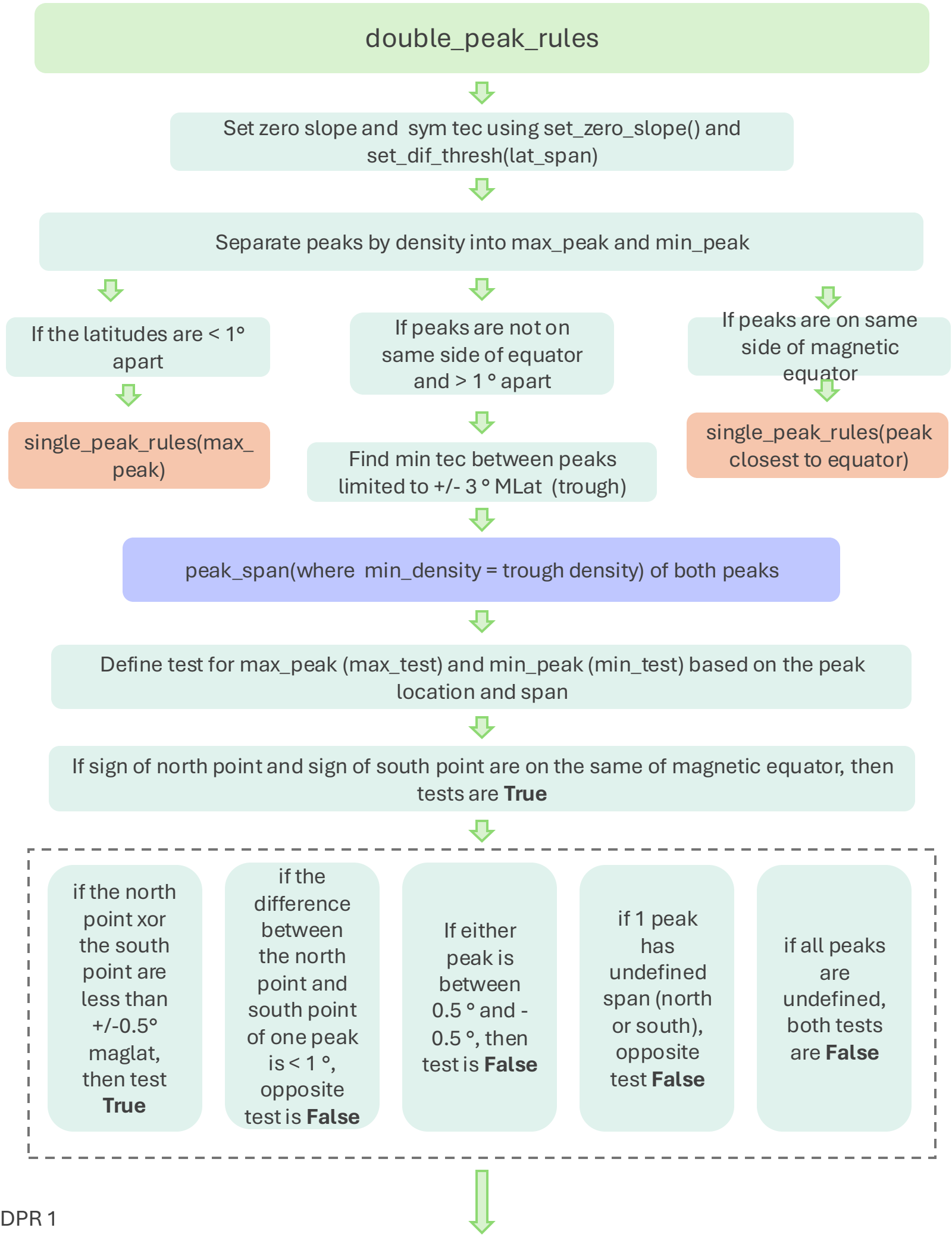

single_peak_rules
peak_span
Fit a line to the density and subtract it from the original Tec to get detrended tec
Using the detrended tec, get the zlopes between -15,0, and 15 maglat
If the the span is undefined and the slopes are – then +
If there is span and slope is not – then +
Trough
flat_rules
If flat == 0
If flat == +/-1
If flat = 2
Peak
Flat (+1 North) (-1 South)
Trough
If lat of peak is > 3, peak north
If lat of peak is < 3 and > -3, just peak
If lat of peak is < -3, peak south

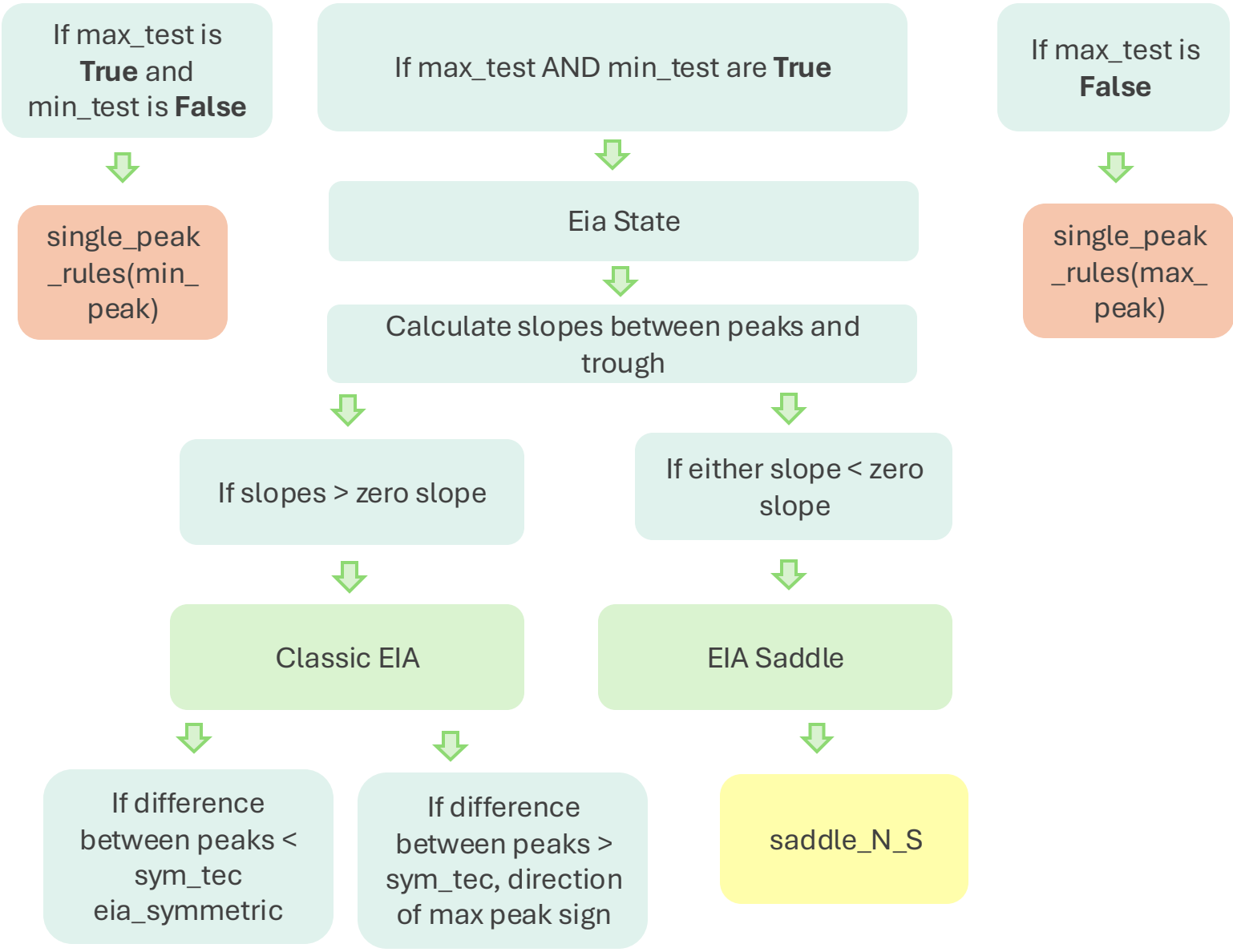

flat_rules
set 0 slope
Find density of zero latitudes
Get density (ρ) north and south of peak (north$_{peak}$ and south$_{peak}$)
Get density (ρ) north (lat > 0) and south (lat<0) (north$_{lat}$ and south$_{lat}$)
set flat to 0
Check 1
North
South
% (ρ north$_{peak}$) > (ρ at peak)
% (ρ south$_{peak}$) > (ρ at peak)
Check 2
North
South
% (ρ north$_{lat}$) < [min(ρ south$_{lat}$)]
% (ρ south$_{lat}$) < [min(ρ north$_{lat}$)]
Check 3
peak_span
North
South
Undefined north point
Undefined south point
Check 4
North
South
Lat at peak within 5° of north edge
Lat at peak within 5° of south edge

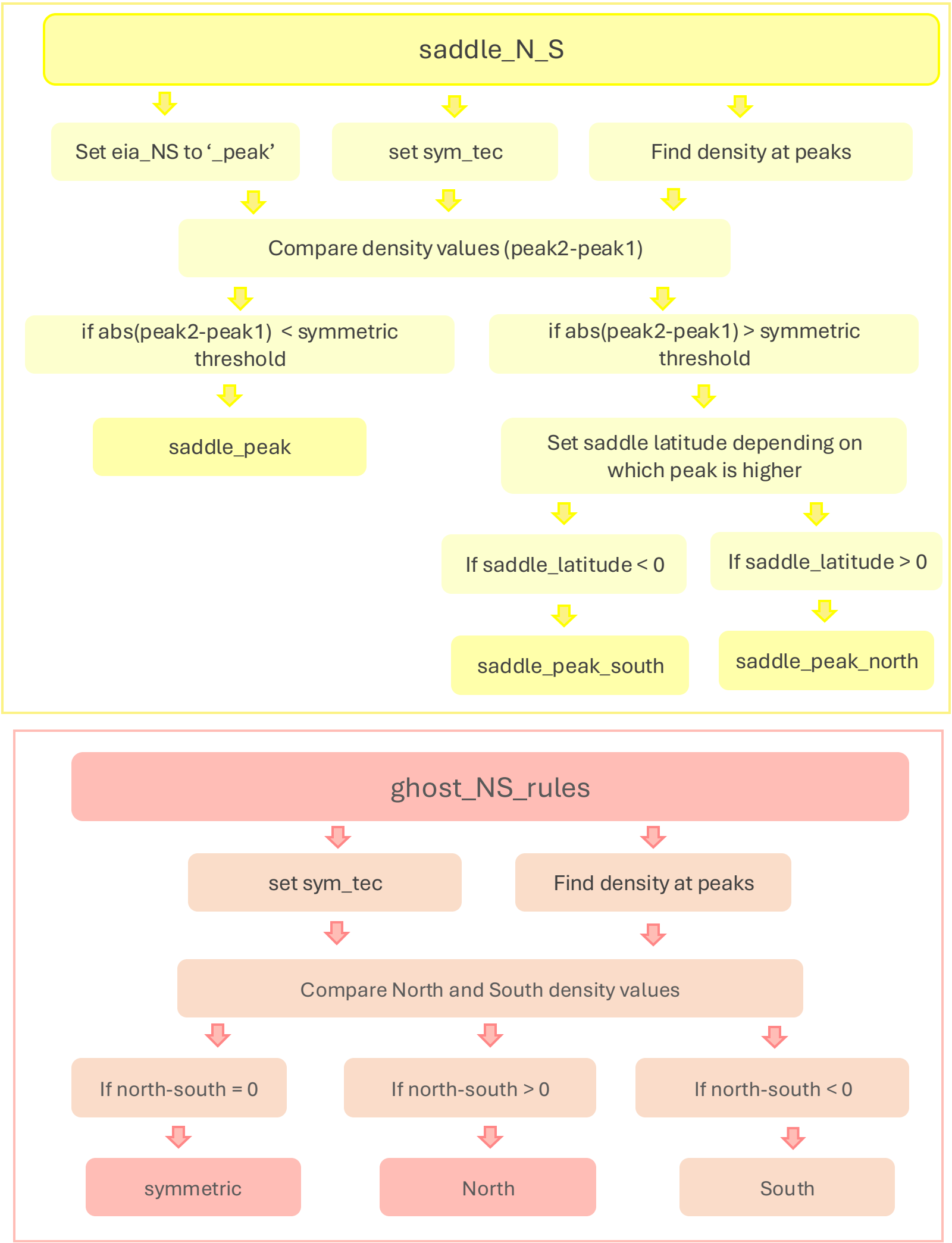
FR 1


Check 1 North > 50 %
xor
Check 1 South > 50 %
set flat to 1
set flat to -1
Check 1 North > 40 %
&
Check 1 South > 40 %
set flat to 2
Check 2 North > 80%
or
Check 2 South > 80%
set flat to -1
set flat to 1
Check 3 North undefined
or
Check 3 South undefined
Check1 North > Check 1 South
Check1 South > Check 1 North
set flat to 1
set flat to -1
Check 4 North edge
or
Check 4 South edge
set flat to 1
set flat to -1
IF flat != 0 AND Check3 North and Check3 South are defined
calculate a linear fit to latitude and density
Subtract fit from original tec to get to get detrended denisty
get_zlopes between south point, peak, and north point of detrended density
IF abs(slope)< zero_slope, slope == 0
If south slope >0 and north slope <0
If south slope = 0 and north slope <0
If south slope > 0 and north slope =0, then NOT FLAT
flat = 0
Return flat (1, -1, 0, 2)

FR 2

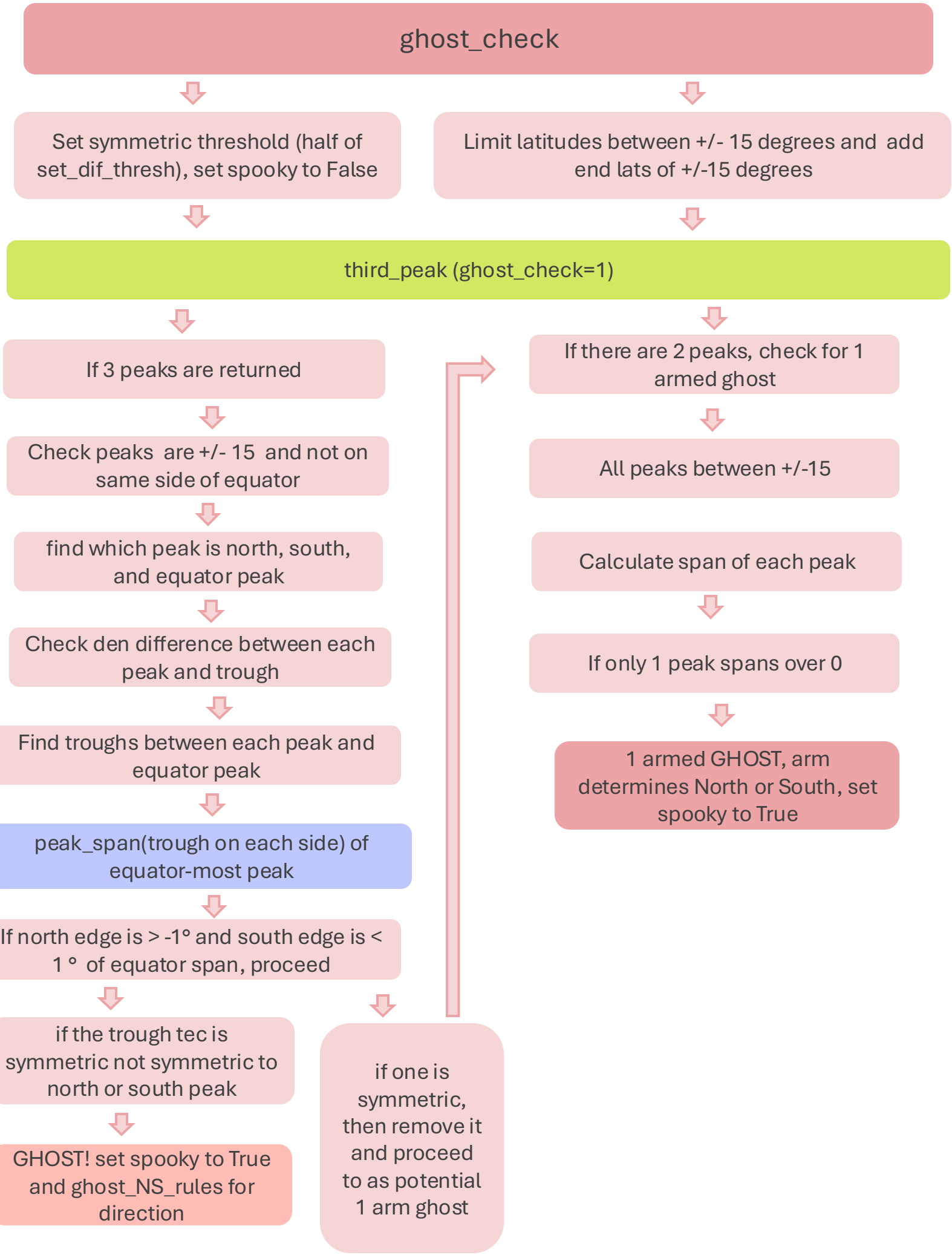

double_peak_rules
Set zero slope and sym tec using set_zero_slope() and set_dif_thresh(lat_span)
Separate peaks by density into max_peak and min_peak
If the latitudes are < 1° apart
single_peak_rules(max_peak)
If peaks are not on same side of equator and > 1 ° apart
Find min tec between peaks limited to +/- 3 ° MLat (trough)
If peaks are on same side of magnetic equator
single_peak_rules(peak closest to equator)
peak_span(where min_density = trough density) of both peaks
Define test for max_peak (max_test) and min_peak (min_test) based on the peak location and span
If sign of north point and sign of south point are on the same of magnetic equator, then tests are True
if the north point xor the south point are less than +/-0.5° maglat, then test True
if the difference between the north point and south point of one peak is < 1 °, opposite test is False
If either peak is between 0.5 ° and -0.5 °, then test is False
if 1 peak has undefined span (north or south), opposite test False
if all peaks are undefined, both tests are False


If max_test is **True** and min_test is **False**
single_peak _rules(min_ peak)
If max_test AND min_test are **True**
Eia State
Calculate slopes between peaks and trough
If slopes > zero slope
If either slope < zero slope
Classic EIA
EIA Saddle
If difference between peaks < sym_tec eia_symmetric
If difference between peaks > sym_tec, direction of max peak sign
saddle_N_S
If max_test is **False**
single_peak _rules(max_ peak)


saddle_N_S
Set eia_NS to '_peak'
set sym_tec
Find density at peaks
Compare density values (peak2-peak1)
if abs(peak2-peak1) < symmetric threshold
if abs(peak2-peak1) > symmetric threshold
saddle_peak
Set saddle latitude depending on which peak is higher
If saddle_latitude < 0
If saddle_latitude > 0
saddle_peak_south
saddle_peak_north
ghost_NS_rules
set sym_tec
Find density at peaks
Compare North and South density values
If north-south = 0
If north-south > 0
If north-south < 0
symmetric
North
South


Peak Span
Find tec at peak lats
Find tec on north and south of the peak
If a trough tec is defined
If a trough tec is not defined
Set t_base to fraction of (peak-trough) + trough
Set t_base to fraction of peak tec
Check for north (south) tec below t_base
Start from peak and search north (south) until a value drops below t_base
Keep track of indices just before tec dropped below t_base, if none found append -99
Loop back until t_base = trough_tec or 1/32 of peak tec
Remove -99 from both north and south indices
If there are non -99 points
If no points are left
Report north and south points of len(points)*div (halfway point if div = 0.5)
Only remove north -99 from north and south -99 from south instead of both
Report defined north and south points using div, else report -99


third_peak
get_zlopes
find_maxima
If ghost_check = 1 AND there are 2 maxima (one arm ghost?)
Get maxima closest and farthest from equator
Check hemisphere of the value farthest from the equator
North hemisphere
South hemisphere
Look for third peak between closest point to equator and -15 mlat
Look for third peak between closest point to equator and +15 mlat
If there are z Locs in the range, choose point farthest from equator
Add a minimum between equator peak and new peak
get_zlopes of new peak
If it is not peak, keep 2 peaks
If it is a peak, add it in to make 3 peaks
If there are is more than 1 max
Return latitudes of maxima
ELSE
If ghost_check = 1
If ghost_check != 1
If there are 3 maxima
If there are not 3 maxima
Return latitudes of maxima
Return empty array


ghost_check
Set symmetric threshold (half of set_dif_thresh), set spooky to False
Limit latitudes between +/- 15 degrees and add end lats of +/-15 degrees
third_peak (ghost_check=1)
If 3 peaks are returned
Check peaks are +/- 15 and not on same side of equator
find which peak is north, south, and equator peak
Check den difference between each peak and trough
Find troughs between each peak and equator peak
peak_span(trough on each side) of equator-most peak
If north edge is > -1° and south edge is < 1 ° of equator span, proceed
if the trough tec is symmetric not symmetric to north or south peak
GHOST! set spooky to True and ghost_NS_rules for direction
if one is symmetric, then remove it and proceed to as potential 1 arm ghost
If there are 2 peaks, check for 1 armed ghost
All peaks between +/-15
Calculate span of each peak
If only 1 peak spans over 0
1 armed GHOST, arm determines North or South, set spooky to True