

Read parameters from config file

rebinned orders and **varRebinned**
PER ORDER, in the header :
start
step
firstUsefulBin
lastUsefulBin

options :
firstOrderToMerge
lastOrderToMerge
default = all orders
in input

step
should have the same value
from **firstOrderToMerge**
until **lastOrderToMerge**

if not, stop with
error message

initialize vectors « **merged** » and « **varMerged** »
with **rebinned.shape[1] * (lastOrderToMerge - firstOrderToMerge + 1)** bins
to the value 0
initialize vectors **startBinOverlap** and **endBinOverlap**
with **lastOrderToMerge - firstOrderToMerge + 1** elements
to the value -1

insert data of **rebinned firstOrderToMerge** in **merged**
starting with **firstUsefulBin [firstOrderToMerge] - 1**
ending with **lastUsefulBin [firstOrderToMerge]**
insert corresponding values of **varRebinned**
in **varMerged**

compute number **startMerged = start[firstOrderToMerge]**
+ step[firstOrderToMerge] * firstUsefulBin [firstOrderToMerge]
set number **stepMerged = step[firstOrderToMerge]**



cycle **kOrder**
over all orders, from
firstOrderToMerge + 1

Take overlapping useful part of both orders
pOrder = kOrder-1

startOverlapP = (start[kOrder] - start[pOrder]) / stepMerged + (firstUsefulBin[kOrder] - 1)
endOverlapP = lastUsefulBin[pOrder] - 1

startOverlapK = firstUsefulBin[kOrder] - 1
endOverlapK = (start[pOrder] - start[kOrder]) / stepMerged + (lastUsefulBin[pOrder] - 1)

There is no common overlap

endOverlapK > startOverlapK

There is a common overlap

extract vector **data** from **rebinned** for **kOrder**
data = rebinned [kOrder]

useData = data [firstUsefulBin[kOrder]-1 : lastUsefulBin[kOrder]]

Find out where to insert **useData** in **merged** :
startOrder = (start [kOrder] - startMerged) / stepMerged + (firstUsefulBin [kOrder] - 1)

merged [startOrder :] = useData

extract vector **varia** from **varRebinned** for **kOrder**
varia = varRebinned [kOrder]

varData = varia [firstUsefulBin[kOrder] -1 : lastUsefulBin[kOrder]]

varMerged [startOrder :] = varData

name both overlapping parts, and size :
overlapP = rebinned [pOrder] [startOverlapP : endOverlapP + 1]
overlapK = rebinned [kOrder] [startOverlapK : endOverlapK + 1]
overlapSize = endOverlapK - startOverlapK + 1
also for error image **varRebinned**

relativePosition = range (overlapSize) / (overlapSize - 1)

overlap = (1.0 - relativePosition) * overlapP + relativePosition * overlapK

varOverlap = (1.0 - relativePosition)^2 * varOverlapP + relativePosition^2 * varOverlapK

startOverlap = (start [kOrder] - startMerged) / stepMerged + (firstUsefulBin [kOrder] - 1)

merged [startOverlap :] = overlap
varMerged [startOverlap :] = varOverlap

startBinOverlap [pOrder] = startOverlap
endBinOverlap [pOrder] = startOverlap + length(overlap) - 1

insert at the right position in merged the part of kOrder following the overlap with pOrder
merged [startOverlap + length (overlap) :] = rebinned [kOrder] [length (overlap) :]

varMerged [startOverlap + length (overlap) :] = varRebinned [kOrder] [length (overlap) :]

Write the results to a FITS file

The FITS image should have a "start" at the value of **startMerged** and its "step" is equal to **stepMerged**

and provide, in addition to the headers already available in the input, for all subsequent orders with a non-empty overlap

HIERARCH MERGE **startOverlapPPP**KKK

HIERARCH MERGE **endOverlapPPP**KKK

where PPP and KKK refer to the two subsequent orders **pOrder** and **kOrder** in format **i3**

from "**firstOrderToMerge** and the next one" up to the "one-but-last and **lastOrderToMerge**" for which the overlap is not empty.

Each of these 'HIERARCH MERGE' entries contains the chosen coordinate (wavelength or ln wavelength) of the bin in '**merged**' where the corresponding overlap starts or ends.