Subject: Re: A better way to find a dip (local minimum with certain conditions)
Posted by Giorgio on Wed, 16 Dec 2009 23:58:47 GMT
View Forum Message <> Reply to Message

On Dec 16, 1:39 pm, chris <rog...@googlemail.com> wrote:
> Hi David,
> as i understood the following might work (untested):
>
> FUNCTION finddip,array,minim,range=range
> range=keyword_set(range)? 0>range<n_elements(array) : 4
> minimum = (min(array-minim,wheremin,/nan))[0] ; [0]-> to get only
> the first match
> if wheremin ne -1 then begin
> ar = array[-1>(wheremin-range):(wheremin+range)<(n_elements
> (array)-1)]
> if max(((arr=ar[sort(ar)]-ar))) eq min(arr) then return,wheremin
> endif else return,-1
> end
>
> Maybe it works and maybe that the principle behind gives you a hint
> for solving the problem
>
> Regards
>
> CR

Hi,
Following CR, this is a small program to find all the local minima.
I did not test it yet.

```
function findip, array,minim
  range = 4
  nel = n_elements(array)
  ; look for candidates
  candidates = where(array GT minim)
  mins = candidates
  ; look which of them is a local minima
  FOR i =0, n_elements(candidates)-1 do begin
    localmin = where(array[((candidates[i]-range)>0):((candidates[i]
+range)<(nel-1) GT array[candidates[i]])
    IF localmin GE 0 THEN mins[i] = 0
  endfor
  return, where(mins NE 0)
end
```