Subject: Re: Help with data gaps and interpolation
Posted by Matthew Argall on Fri, 20 Mar 2015 16:34:08 GMT
View Forum Message <> Reply to Message

There is a lot going wrong in this example.

```
   index = where(Bx_array eq 9999.99, count)
    if (count gt 9999.99) then Bx_array[index] = !values.f_nan
```

"count" is an integer telling you how man fill values you have. It should not be compared to your fill value.

```
if Bx_array lt 9999.99 then begin
   temp=(Bx_array, count)
   result=interpol(Bx_array[temp],temp, time[temp])
 endif
```

Bx_array is an array and comparing it to a scalar (9999.99) will cause an error. The line temp = (Bx_array, count) will append the scalar integer "count" to the end of the Bx_array.
Bx_array[temp] indexes an array with another array. This would be ok if "temp" were an array of indices, but it is not. Thus, your call to "interpol" does not give you what you expect.

Something like this should work:

```
;Create data with fill values
Bx = randomu(3, 50)
Bx[[3, 4, 5, 12, 30, 33, 39, 45]] = 9999.99

;Find the good index values
iGood = where(Bx ne 9999.99, nGood)

;Create a time array indicating where you want points
nPts = n_elements(Bx)
time_out = findgen(nPts)

;Times where you have good data
time_in = time_out[iGood]
Bx_in    = Bx[iGood]

;Interpolate over fill values
Bx_out = Interpol(Bx_in, time_in, time_out)
```