"Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message
news:10bcckpsje3st3e@corp.supernews.com...
>
> "Ryan" <rab209@hotmail.com> wrote in message
> news:f845cc9f.0405270943.1a586412@posting.google.com...
>> Hi all,
>>
>> I'm using the PARTICLE_TRACE procedure to simulate a particle flow
>> through a 3-D velocity field, which, in most cases, works well.
>>
>> However, on the rare occasion where the seeds don't go anywhere
>> (usually due to the velocity vectors immediately pushing the seeds out
>> of the simulation volume), the program crashes, and IDL tells me that
>>
>> % Array dimensions must be greater than 0.
>>
>> and it points me to the line where my PARTICLE_TRACE call is made.
>> When examining the output values after the crash, the "Verts" and
>> "Normals" arrays are fine, but the "Conn" parameter is undefined...
>>
>> Which makes me think that PARTICLE_TRACE needs to have at least one
>> connecting line for it to work w/o crashing... which is unfortunate,
>> since I can't .cont my way back afterwards, nor can I automatically
>> predict if that would happen beforehand by looking at the data.
>>
>> Any ideas on if this is what's really happening and/or a workaround?
>>
>> Thanks much,
>> Ryan
>
> You are right about it not working when there are no lines to return.
> This has already been found and fixed for IDL 6.1.
> In 6.1, when there are no lines to return, the conn array is returned with
> the contents [-1].
> I'm sorry that I can't think of a good workaround.

I don't know why, but sometimes the answer pops up in my head a day later.

Use a CATCH block:

```
pro pt
   data = FLTARR(2, 500, 500)
   data[1,*,*] = 100000
   seeds = [[500,500]]
```

```
   catch, err
   if err ne 0 then begin
      print, "In catch"
      help, verts
      help, conn
      conn = [-1]
      catch, /cancel
   endif $
   else $
      PARTICLE_TRACE, data, seeds, verts, conn
   help, verts
   help, conn
end
```