Subject: Re: reading large multicolumn data file
Posted by penteado on Fri, 06 Aug 2010 03:05:14 GMT
View Forum Message <> Reply to Message

On Aug 5, 8:54 pm, bio_amateur <hoangtrongminht...@gmail.com> wrote:
> I have a data file (a few hundreds MB). This is a text file in the
> format X Y1 Y2 Y3 (first column is the common x-axis data, next
> columns are data). I can read the data and plot easily with xmgrace
> using
>
> xmgrace -nxy data.dat
>
> which take a few seconds to plot. Now, I want to use IDL to read this
> file and display using iTool. What I did was
>
> data = read_ascii(filename)
> myPlotData = data.(0)
> rows = (size(myPlotData, /dimension))[0]
> for ii=1, rows do begin
> iPlot, myPlotData[0,*], myPlotData[ii,*], /overplot
> end
>
> This method takes so long. Could someone good at this can point out a
> solution for me.

The iTools, as any use of object graphics, can get heavy in memory
when the number of vertices is large (typically, from several hundred
thousand). This may be alleviated if change your IDL preferences to
use hardware rendering, in case it is not already in hardware. Other
than that, only using direct graphics will solve it, as in

```
minx=min(myplotdata[0,*],max=maxx)
miny=min(myplotdata[1:*,*],max=maxy)

plot,[minx,maxx],[miny,maxy],/nodata
for ii=1,rows-1 do oplot,myplotdata[0,*],myplotdata[ii,*]
```

Subject: Re: reading large multicolumn data file
Posted by bio_amateur on Fri, 06 Aug 2010 14:23:45 GMT
View Forum Message <> Reply to Message

On Aug 5, 11:05 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Aug 5, 8:54 pm, bio_amateur <hoangtrongminht...@gmail.com> wrote:
>
>
>
>> I have a data file (a few hundreds MB). This is a text file in the

>> format X Y1 Y2 Y3 (first column is the common x-axis data, next
>> columns are data). I can read the data and plot easily with xmgrace
>> using
>
>> xmgrace -nxy data.dat
>
>> which take a few seconds to plot. Now, I want to use IDL to read this
>> file and display using iTool. What I did was
>
>> data = read_ascii(filename)
>> myPlotData = data.(0)
>> rows = (size(myPlotData, /dimension))[0]
>> for ii=1, rows do begin
>> iPlot, myPlotData[0,*], myPlotData[ii,*], /overplot
>> end
>
>> This method takes so long. Could someone good at this can point out a
>> solution for me.
>
> The iTools, as any use of object graphics, can get heavy in memory
> when the number of vertices is large (typically, from several hundred
> thousand). This may be alleviated if change your IDL preferences to
> use hardware rendering, in case it is not already in hardware. Other
> than that, only using direct graphics will solve it, as in
>
> minx=min(myplotdata[0,*],max=maxx)
> miny=min(myplotdata[1:*,*],max=maxy)
>
> plot,[minx,maxx],[miny,maxy],/nodata
> for ii=1,rows-1 do oplot,myplotdata[0,*],myplotdata[ii,*]

Now, plotting is pretty fast. However, data reading is still slow,
compared to xmgrace. Is there a better way than using the 2 steps:
read the data into a structure, and convert a structure to an array as
the way I did.

The second question is xmgrace autoamtically choose one color for each
data field (Y1, Y2, ...). However, with direct graphics in IDL, I only
get one single color. How could we resolve this in IDL.

Thanks,
Tuan

Subject: Re: reading large multicolumn data file
Posted by bio_amateur on Fri, 06 Aug 2010 14:23:58 GMT
View Forum Message <> Reply to Message

On Aug 5, 11:05 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Aug 5, 8:54 pm, bio_amateur <hoangtrongminht...@gmail.com> wrote:
>
>
>
>> I have a data file (a few hundreds MB). This is a text file in the
>> format X Y1 Y2 Y3 (first column is the common x-axis data, next
>> columns are data). I can read the data and plot easily with xmgrace
>> using
>
>> xmgrace -nxy data.dat
>
>> which take a few seconds to plot. Now, I want to use IDL to read this
>> file and display using iTool. What I did was
>
>> data = read_ascii(filename)
>> myPlotData = data.(0)
>> rows = (size(myPlotData, /dimension))[0]
>> for ii=1, rows do begin
>> iPlot, myPlotData[0,*], myPlotData[ii,*], /overplot
>> end
>
>> This method takes so long. Could someone good at this can point out a
>> solution for me.
>
> The iTools, as any use of object graphics, can get heavy in memory
> when the number of vertices is large (typically, from several hundred
> thousand). This may be alleviated if change your IDL preferences to
> use hardware rendering, in case it is not already in hardware. Other
> than that, only using direct graphics will solve it, as in
>
> minx=min(myplotdata[0,*],max=maxx)
> miny=min(myplotdata[1:*,*],max=maxy)
>
> plot,[minx,maxx],[miny,maxy],/nodata
> for ii=1,rows-1 do oplot,myplotdata[0,*],myplotdata[ii,*]

Now, plotting is pretty fast. However, data reading is still slow,
compared to xmgrace. Is there a better way than using the 2 steps:
read the data into a structure, and convert a structure to an array as
the way I did.

The second question is xmgrace autoamtically choose one color for each
data field (Y1, Y2, ...). However, with direct graphics in IDL, I only
get one single color. How could we resolve this in IDL.

Thanks,
Tuan

Subject: Re: reading large multicolumn data file
Posted by David Fanning on Fri, 06 Aug 2010 14:46:21 GMT
View Forum Message <> Reply to Message

bio_amateur writes:

> Now, plotting is pretty fast. However, data reading is still slow,
> compared to xmgrace. Is there a better way than using the 2 steps:
> read the data into a structure, and convert a structure to an array as
> the way I did.
>
> The second question is xmgrace autoamtically choose one color for each
> data field (Y1, Y2, ...). However, with direct graphics in IDL, I only
> get one single color. How could we resolve this in IDL.

I'd try something like this:

```
rows = File_Lines(filename)
data = FltArr(4, rows)
Openr, lun, filename, /Get_Lun
ReadF, lun, data
Free_Lun, data
data = Transpose(data)

colors = ['black', 'white', 'red', 'green', 'blue']
Plot, data[*,0], data[1,*], Background=FSC_Color(colors[1]), $
   Color=FSC_Color(colors[0]), /NoData
FOR j=1,3 DO OPlot, data[*,0], data[*,j], COLOR=FSC_Color(colors[j+1])
```

You can find FSC_Color in the Coyote Library:

   http://www.dfanning.com/documents/programs.html

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: reading large multicolumn data file
Posted by David Fanning on Fri, 06 Aug 2010 15:17:53 GMT

View Forum Message <> Reply to Message

David Fanning writes:

> Free_Lun, data

Whoops! That's not likely to work. :-(

Try, "Free_Lun, lun"

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: reading large multicolumn data file
Posted by bio_amateur on Fri, 06 Aug 2010 18:22:12 GMT
View Forum Message <> Reply to Message

On Aug 6, 11:17 am, David Fanning <n...@dfanning.com> wrote:
> David Fanning writes:
>> Free_Lun, data
>
> Whoops! That's not likely to work. :-(
>
> Try, "Free_Lun, lun"
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

It works. Many thanks for this powerful package, David.

Tuan.
