
Subject: read an irregularly structured ascii file.

Posted by [Russell Ryan](#) on Tue, 25 Sep 2012 18:35:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi gang,

I'm usually pretty good at these sorts of things, but this one is giving me all sorts of grief. I have a host of ascii files (each is about 60k lines) that contain several scalars and multi-dimensional arrays. The files were created with Fortran in a very specific manner, which seems fairly straightforward, but reading them into IDL has been a pain. Basically, the file is like this:

```
221 0. 125893. 141254. 158489. 177828. 199527. 223873. 251189.
281839. 316229. 354815. 398109. 446686. 501190. 562344. 630961.
707950. 794334. 891257. 1000000. 1047130. 1096480. 1148150. 1202260.
1258930. 1318260. 1380380. 1445440. 1513560. 1584890. 1659590.
1737800. 1819700. 1905460. 1995260. 2089290. 2187760. 2290870.
2398830. 2511880. 2630270. 2754230. 2884030. 3019950. 3162270.
....
1036831949 1120403456 0
1. 1.49025571 0.671025753 0 0. 0. 0. 1. 1. -2 F
SOME ASCII TEXT HERE
SOME ASCII TEXT HERE
SOME ASCII TEXT HERE
1221 91. 94. 96. 98. 100. 102. 104. 106. 108. 110. 114. 118. 121.
125. 127. 128. 131. 132. 134. 137. 140. 143. 147. 151. 155. 159.
162. 166. 170. 173. 177. 180. 182. 186. 191. 194. 198. 202. 205.
210. 216. 220. 223. 227. 230. 234. 240. 246. 252. 257. 260. 264.
269. 274. 279. 284. 290. 296. 301. 308. 318. 328. 338. 348. 357.
366. 375. 385. 395. 405. 414. 422. 430. 441. 451. 460. 470. 480.
490. 500. 506. 512. 520. 530. 540. 550. 560. 570. 580. 590. 600.
610. 620. 630. 640. 650. 658. 665. 675. 685. 695. 705. 716. 726.
...
1221 2.64753693E-18 8.0283098E-18 2.30857457E-17 4.34018954E-17
8.86453794E-17 2.1266836E-16 3.95873712E-16 7.21644172E-16 1.42709561E-15
3.33774985E-15 9.49530287E-15 2.67932361E-14 6.68887688E-14
1.55580545E-13 2.66895013E-13 3.59504581E-13 5.61764719E-13
8.51162456E-13 1.15773938E-12 1.97114634E-12 3.71074482E-12
7.07988225E-12 1.35896346E-11 2.64187925E-11 5.14091755E-11
...
```

To explain, the first entry (is an integer) which specifies the number of elements the next array contains. So, the 221 means a floating-point array of 221 elements is coming next (that array is truncated and represented by ... in my example). Then there is some jibberish, as the two long numbers: 1036831949 1120403456 0. Then there is a line of useful information: 1. 1.49025571 0.671025753 0 0. 0. 0. 1. 1. -2 F. Then there are three lines in plain text (which are useful also). Then it begins with the vectors again. Like before, the first number is an integer which indicates the number of elements which are coming. So in the first case, there are 1221 elements

in the next array. This pattern of 1221 long arrays will then repeat for 221 times.

Obviously, I can read the entire dataset into memory as a string, then do several very tedious string operations to sort out the data. But that is, tedious. I was hoping to use read/readf/reads like you might with Fortran. Any ideas?

Thanks a million!
Russell
