readCSV

Neal Nesbitt <NesbittN6830@uhcl.edu>

May 05 2016

Abstract

This function takes as a parameter the directory location of a comma separated value file and reads it into memory. It then calculates the maximum and minimum value for each column and displays it to screen.

1 Introduction

This is one of a suite of functions in development for implementing I/O with comma separated value (csv) files. They are intended to allow easy portability of scalar and vector points in the global computational grid into and out of human readable .csv files as used by programs such as Microsoft Excel and MATLAB.

2 Physical System

The points in question to be read or written are considered either scalar or vector quantities. Currently readCSV will read any csv with constant column length into memory, but it does not translate that into a grid variable. The next section of code to be added will compute the minimum spacing between each coordinate (column) of the points (rows) given, and use half of this value as the spacing for the computational grid. Currently it only computes the minimums and maximums for each coordinate, which will be used to specify the grid boundaries.

3 Numerical Implementation

The program is currently a single section of code meant to stream through the data and read the values as floating point numbers into memory. This is done character by character with flagged procedures for special delimiters. Because of this, the file can be as large as memory allows, both in the number of rows and columns. The only constraint is that each row must have the same number of columns.

4 Using This Thorn

The parameter file must contain the variable "`filePath`" that holds a string with the file path of the csv file to be read in.

4.1 Basic Usage

Depending on the directory location of the csv file, the parameter file only needs the lines:

```
ActiveThorns = "readCSV"
readCSV::filePath = "/home/myUserName/csvFiles/myCSVFile.csv"
```

For this example my login name for the cluster is `myUserName`, and the file `myCSVFile.csv` is in a folder in my personal directory called `csvFiles/`

4.2 Special Behaviour

The function should throw errors and abort if:

- the file doesn't open or is corrupted,
- the rows have different numbers of columns,
- the entries don't represent numbers in floating point format,
- or if any of the internal buffers fail to resize properly.