Simple Plain Text Output with IOASCII

Thomas Radke, Gabrielle Allen

Date: 2004/03/13 18:26:42

Abstract

Thorn **IOASCII** provides I/O methods for 1D, 2D, and 3D output of grid arrays and grid functions into files in ASCII format. The precise format is designed for visualisation using the clients `xgraph` [1] or `gnuplot` [2].

1 Purpose

Thorn **IOASCII** registers three I/O methods named `IOASCII_1D`, `IOASCII_2D`, and `IOASCII_3D` with the I/O interface in the flesh.

- `IOASCII_1D`
  creates one-dimensional output of 1D, 2D and 3D grid functions and arrays by slicing through the edge (in the octant case) or center (in all origin centered cases) of the grid in the coordinate directions. In addition, output is provided along a diagonal of the grid, in this case the diagonal always starts at the first grid point (that is, in Fortran notation `var(1,1,1)`) and the line taken uses grid points increasing by 1 in each direction. [NOTE: The diagonal output is not available for staggered variables].
  Output for each direction can be selected individually via parameters.
  Data is written in ASCII format and goes into files with the names:

  `<variable_name>_<slice>_[<center_i>][<center_j>].{asc|xg}`

  and for diagonals:

  `<variable_name>_3D_diagonal.{asc|xg}`

  These files can be processed directly by either xgraph or gnuplot (you can select the style of output via parameter settings).

- `IOASCII_2D` outputs two-dimensional slices of grid functions and arrays planes. Again, slicing is done through the edge (in the octant case) or center (in all origin centered cases).
  Data is written in ASCII format and goes into files named

  `<variable_name>_<plane>_[<center>].{asc|xg}`

  These files can be visualized by gnuplot using its *splot* command.

- `IOASCII_3D` outputs three-dimensional grid functions and arrays as a whole.
  Data is written in ASCII format and goes into files named

  `<variable_name>_3D.{asc|xg}`

  These files can be visualized by gnuplot using its *splot* command.

2 IOASCII Parameters

General parameters to control all **IOASCII**'s I/O methods are:

- `IOASCII::out[123]D_every` (steerable)
  How often to do periodic `IOASCII` output. If this parameter is set in the parameter file, it will override the setting of the shared `IO::out_every` parameter. The output frequency can also be set for individual variables using the `out_every` option in an option string appended to the `IOASCII::out[123]D_vars` parameter.

- `IOASCII::out[123]D_vars` (steerable)
  The list of variables to output using an **IOASCII** I/O method. The variables must be given by their fully qualified variable or group name. The special keyword *all* requests `IOASCII` output for all variables. Multiple names must be separated by whitespaces.
  An option string can be appended in curly braces to the group/variable name. The only option supported so far is `out_every` which sets the output frequency for an individual variable (overriding `IOASCII::out[123]D_every` and `IO::out_every`).

- `IOASCII::out[123]D_dir`
  The directory in which to place the `IOASCII` output files. If the directory doesn't exist at startup it will be created.
  If this parameter is set to an empty string `IOASCII` output will go to the standard output directory as specified in `IO::out_dir`.

- `IOASCII::out_format` (steerable)
  The output format for floating-point numbers in `IOASCII` output.
  This parameter conforms to the format modifier of the C library routine *fprintf(3)*. You can set the format for outputting floating-point numbers (fixed or exponential) as well as their precision (number of digits).

- `IOASCII::out[123]D_style`
  The output style for `IOASCII` output.
  This parameter chooses between `gnuplot`- and `xgraph`-suitable output style, and – for `gnuplot` – determines whether to also plot the physical time in the output data or not.

Additional parameters to control the `IOASCII_1D` I/O method are:

- `IOASCII::out1D_[xyzd]` (steerable)
  Chooses the directions to output in 1D `IOASCII` format (`d` stands for diagonal direction).

- `IOASCII::out1D_[xyz]line_[xyz], IOASCII::out1D_[xyz]line_[xyz]i`
  Chooses the slice centers for 1D lines from `IOASCII_1D`. These can be set either in physical or index coordinates. If they are set these parameters will override the default slice center parameters `IO::out_[xyz]line[xyz], IO::out_[xyz]line[xyz]i`. Note that the slice center can only be set for grid functions so far. For `CCTK_ARRAY` variables the slices will always start in the origin, ie. (0, 0, 0) in C ordering.

Additional parameters to control the `IOASCII_2D` I/O method are:

- `IOASCII::out2D_[xyxzyz]plane_[xyz], IOASCII::out2D_[xyxzyz]plane_[xyz]i`
  Chooses the slice centers for 2D planes from `IOASCII_2D`. These can be set either in physical or index coordinates. If they are set these parameters will override the default slice center parameters `IO::out_[xyxzyz]plane[xyz], IO::out_[xyxzyz]plane[xyz]i`.

3 Comments

Getting Output from IOBasic's I/O Mehtods
You obtain output by an I/O method by either

- setting the appropriate I/O parameters
- calling one the routines of the I/O function interface provided by the flesh

For a description of basic I/O parameters and the I/O function interface to invoke I/O methods by application thorns please see the documentation of thorn **IOUtil** and the flesh.

Building Cactus configurations with IOBasic
Since **IOASCII** uses parameters from **IOUtil** it also needs this I/O helper thorn be compiled into Cactus and activated at runtime in the `ActiveThorns` parameter in your parameter file.

4 Examples

In this section we include example output for different parameter combinations. Note that all these examples were generated for just a couple of timesteps for an extremely small 3D grid.

4.1 One-dimensional xgraph

These options produce data suitable for using with the xgraph visualization client in the format `x f(t=fixed,x,y=fixed,z=fixed)`:

```
IOASCII::out1D_every = 1
IOASCII::out1D_vars = "wavetoy::phi"
IOASCII::out1D_style = "xgraph"
```

Output File: phi_x_[1][1].xg

```
"Parameter file wavetoy.par
"Created Sun 19 Aug 2001 16:31:43
"x-label x
"y-label WAVETOY::phi (y = 0.0000000000000, z = 0.0000000000000), (yi = 1, zi = 1)


"Time = 0.0000000000000
-0.5000000000000 0.0000000000139
0.0000000000000 1.0000000000000
0.5000000000000 0.0000000000139


"Time = 0.2500000000000
-0.5000000000000 0.0000000000000
0.0000000000000 0.4980695458846
0.5000000000000 0.0000000000000


"Time = 0.5000000000000
-0.5000000000000 0.0019304541362
0.0000000000000 -0.7509652270577
0.5000000000000 0.0019304541362
```

4.2 One-dimensional gnuplot

These options produce data suitable for using with the gnuplot visualization client in the format `x f(t,x,y=fixed,z=fixed)`:

```
IOASCII::out1D_every = 1
IOASCII::out1D_vars = "wavetoy::phi"
IOASCII::out1D_style = "gnuplot f(x)"
```

Output File: phi_x_[1][1].asc

```
#Parameter file wavetoy.par
#Created Sun 19 Aug 2001 16:33:07
#x-label x
#y-label WAVETOY::phi (y = 0.0000000000000, z = 0.0000000000000), (yi = 1, zi = 1)

#Time = 0.0000000000000
-0.5000000000000 0.0000000000139
0.0000000000000 1.0000000000000
0.5000000000000 0.0000000000139

#Time = 0.2500000000000
-0.5000000000000 0.0000000000000
0.0000000000000 0.4980695458846
0.5000000000000 0.0000000000000

#Time = 0.5000000000000
-0.5000000000000 0.0019304541362
0.0000000000000 -0.7509652270577
0.5000000000000 0.0019304541362
```

4.3 One-dimensional gnuplot (including time)

These options produce data suitable for using with the gnuplot visualization client in the format `t x f(t,x,y=fixed,z=fixed)`:

```
IOASCII::out1D_every = 1
IOASCII::out1D_vars = "wavetoy::phi"
IOASCII::out1D_style = "gnuplot f(t,x)"
```

Output file: phi_x_[1][1].asc

```
#Parameter file wavetoy.par
#Created Sun 19 Aug 2001 16:34:48
#x-label x
#y-label WAVETOY::phi (y = 0.0000000000000, z = 0.0000000000000), (yi = 1, zi = 1)

#Time = 0.0000000000000
0.0000000000000 -0.5000000000000 0.0000000000139
0.0000000000000 0.0000000000000 1.0000000000000
0.0000000000000 0.5000000000000 0.0000000000139

#Time = 0.2500000000000
0.2500000000000 -0.5000000000000 0.0000000000000
0.2500000000000 0.0000000000000 0.4980695458846
0.2500000000000 0.5000000000000 0.0000000000000

#Time = 0.5000000000000
0.5000000000000 -0.5000000000000 0.0019304541362
0.5000000000000 0.0000000000000 -0.7509652270577
0.5000000000000 0.5000000000000 0.0019304541362
```

4.4 Two-dimensional gnuplot

These options produce data suitable for using with the gnuplot visualization client in the format `x y f(t,x,y,z=fixed)`:

```
IOASCII::out2D_every = 1
IOASCII::out2D_vars = "wavetoy::phi"
IOASCII::out2D_style = "gnuplot f(x,y)"
```

Output file: phi_xy_[1].asc

```
#Parameter file wavetoy.par
#Created Sun 19 Aug 2001 16:31:43
#x-label x
#y-label y
#z-label WAVETOY::phi (z = 0.0000000000000), (zi = 1)


#Time = 0.0000000000000
-0.5000000000000 -0.5000000000000 0.0000000000000
0.0000000000000 -0.5000000000000 0.0000000000139
0.5000000000000 -0.5000000000000 0.0000000000000

-0.5000000000000 0.0000000000000 0.0000000000139
0.0000000000000 0.0000000000000 1.0000000000000
0.5000000000000 0.0000000000000 0.0000000000139

-0.5000000000000 0.5000000000000 0.0000000000000
0.0000000000000 0.5000000000000 0.0000000000139
0.5000000000000 0.5000000000000 0.0000000000000



#Time = 0.2500000000000
-0.5000000000000 -0.5000000000000 0.0000000000000
0.0000000000000 -0.5000000000000 0.0000000000000
0.5000000000000 -0.5000000000000 0.0000000000000

-0.5000000000000 0.0000000000000 0.0000000000000
0.0000000000000 0.0000000000000 0.4980695458846
0.5000000000000 0.0000000000000 0.0000000000000

-0.5000000000000 0.5000000000000 0.0000000000000
0.0000000000000 0.5000000000000 0.0000000000000
0.5000000000000 0.5000000000000 0.0000000000000



#Time = 0.5000000000000
-0.5000000000000 -0.5000000000000 0.0000000008425
0.0000000000000 -0.5000000000000 0.0019304541362
0.5000000000000 -0.5000000000000 0.0000000008425

-0.5000000000000 0.0000000000000 0.0019304541362
0.0000000000000 0.0000000000000 -0.7509652270577
0.5000000000000 0.0000000000000 0.0019304541362

-0.5000000000000 0.5000000000000 0.0000000008425
0.0000000000000 0.5000000000000 0.0019304541362
0.5000000000000 0.5000000000000 0.0000000008425
```

4.5 Two-dimensional gnuplot (including time)

These options produce data suitable for using with the gnuplot visualization client in the format `t x y f(t,x,y,z=fixed)`:

```
IOASCII::out2D_every = 1
IOASCII::out2D_vars = "wavetoy::phi"
IOASCII::out2D_style = "gnuplot f(t,x,y)"
```

Output file: phi_xy_[1].asc

```
#Parameter file wavetoy.par
#Created Sun 19 Aug 2001 16:33:07
#x-label x
#y-label y
#z-label WAVETOY::phi (z = 0.0000000000000), (zi = 1)


#Time = 0.0000000000000
0.0000000000000 -0.5000000000000 -0.5000000000000 0.0000000000000
0.0000000000000 0.0000000000000 -0.5000000000000 0.0000000000139
0.0000000000000 0.5000000000000 -0.5000000000000 0.0000000000000

0.0000000000000 -0.5000000000000 0.0000000000000 0.0000000000139
0.0000000000000 0.0000000000000 0.0000000000000 1.0000000000000
0.0000000000000 0.5000000000000 0.0000000000000 0.0000000000139

0.0000000000000 -0.5000000000000 0.5000000000000 0.0000000000000
0.0000000000000 0.0000000000000 0.5000000000000 0.0000000000139
0.0000000000000 0.5000000000000 0.5000000000000 0.0000000000000



#Time = 0.2500000000000
0.2500000000000 -0.5000000000000 -0.5000000000000 0.0000000000000
0.2500000000000 0.0000000000000 -0.5000000000000 0.0000000000000
0.2500000000000 0.5000000000000 -0.5000000000000 0.0000000000000

0.2500000000000 -0.5000000000000 0.0000000000000 0.0000000000000
0.2500000000000 0.0000000000000 0.0000000000000 0.4980695458846
0.2500000000000 0.5000000000000 0.0000000000000 0.0000000000000

0.2500000000000 -0.5000000000000 0.5000000000000 0.0000000000000
0.2500000000000 0.0000000000000 0.5000000000000 0.0000000000000
0.2500000000000 0.5000000000000 0.5000000000000 0.0000000000000



#Time = 0.5000000000000
0.5000000000000 -0.5000000000000 -0.5000000000000 0.0000000008425
0.5000000000000 0.0000000000000 -0.5000000000000 0.0019304541362
0.5000000000000 0.5000000000000 -0.5000000000000 0.0000000008425

0.5000000000000 -0.5000000000000 0.0000000000000 0.0019304541362
0.5000000000000 0.0000000000000 0.0000000000000 -0.7509652270577
0.5000000000000 0.5000000000000 0.0000000000000 0.0019304541362
```

```
0.5000000000000 -0.5000000000000 0.5000000000000 0.0000000008425
0.5000000000000 0.0000000000000 0.5000000000000 0.0019304541362
0.5000000000000 0.5000000000000 0.5000000000000 0.0000000008425
```

References

[1] http://www.cactuscode.org/VizTools/xgraph.html, http://jean-luc.aei.mpg.de/Codes/xgraph/

[2] http://www.cactuscode.org/VizTools/Gnuplot.html, http://www.gnuplot.info