IOBasic

Gabrielle Allen
Thomas Radke

Date: 2004/05/06 09:05:56

Abstract

Thorn **IOBasic** provides I/O methods for outputting scalar values in ASCII format into files and for printing them as runtime information to screen.

1 Purpose

Thorn **IOBasic** registers 2 I/O methods with the I/O interface in the flesh which both output the same following information:

- for CCTK_SCALAR variables, the value of the scalar versus time
- for CCTK_GF and CCTK_ARRAY variables, the values of global reduction operations (eg. minimum, maximum, L1, and L2 norm) versus time

The I/O methods differ in the destination the output is written to:

- `Scalar`
  This method outputs the information into ASCII files named `"<scalar_name>.{asc|xg}"` (for CCTK_SCALAR variables) and `"<var_name>_<reduction>.{asc|xg}"` (for CCTK_GF and CCTK_ARRAY variables where `reduction` would stand for the type of reduction value that is output).
  The output data can be plotted by using either *xgraph* (for `"*.xg"` files) or *gnuplot* (for `"*.asc"` files). The output style can be selected via parameter settings.
- `Info`
  This method prints the data as runtime information to *stdout*.
  The output occurs as a table with columns containing the current iteration number, the physical time at this iteration, and more columns for scalar/reduction values of each variable to be output.

2 IOBasic Parameters

Parameters to control the `Scalar` I/O method are:

- `IOBasic::outScalar_criterion` (steerable)
  The criterion that decides when to `Scalar` output. If this parameter is set in the parameter file, it will override the setting of the shared `IO::out_criterion` parameter.
- `IOBasic::outScalar_every` (steerable)
  How often, in terms of iterations, to do `Scalar` output. If this parameter is set in the parameter file, it will override the setting of the shared `IO::out_every` parameter.
- `IOBasic::outScalar_dt` (steerable)
  How often, in terms of simulation time, to do `Scalar` output. If this parameter is set in the parameter file, it will override the setting of the shared `IO::out_dt` parameter.

- `IOBasic::out_dir`
  The directory in which to place the `Scalar` ASCII output files. If the directory doesn't exist at startup it will be created.
  If this paraeter is set to an empty string `Scalar` output will go to the standard output directory as specified in `IO::out_dir`.

- `IOBasic::outScalar_style`
  How to start comments in the `Scalar` ASCII output files.
  Possible choices for this keywork parameter are *xgraph* and *gnuplot*.

- `IOBasic::out_format` (steerable)
  The output format for floating-point numbers in `Scalar` 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).

- `IOBasic::outScalar_vars` (steerable)
  The list of variables to output into individual ASCII files.
  The variables must be given by their fully qualified variable or group name. The special keyword *all* requests `Scalar` output for all variables. Multiple variables must be separated by spaces. For CCTK_GF and CCTK_ARRAY variables, an option string can be appended in curly braces to the name of the variable. The only option supported so far is an individual list of reductions for that variable which would take precedence over the default reduction operations to perform.

- `IOBasic::outScalar_reductions` (steerable)
  The list of global reduction operations to perform on CCTK_GF and CCTK_ARRAY variables for `Scalar` output. This setting can be overridden for individual variables using an option string.
  Multiple reduction names must be separated by spaces.

Parameters to control the `Info` I/O method are:

- `IOBasic::outInfo_criterion` (steerable)
  The criterion that decides when to `Info` output. If this parameter is set in the parameter file, it will override the setting of the shared `IO::out_criterion` parameter.

- `IOBasic::outInfo_every` (steerable)
  How often, in terms of iterations, to do `Info` output. If this parameter is set in the parameter file, it will override the setting of the shared `IO::out_every` parameter.

- `IOBasic::outInfo_dt` (steerable)
  How often, in terms of simulation time, to do `Info` output. If this parameter is set in the parameter file, it will override the setting of the shared `IO::out_dt` parameter.

- `IOBasic::outInfo_vars` (steerable)
  The list of variables to output to screen.
  The variables must be given by their fully qualified variable or group name. The special keyword *all* requests `Info` output for all variables. Multiple variables must be separated by spaces.
  For CCTK_GF and CCTK_ARRAY variables, an option string can be appended in curly braces to the name of the variable. The only option supported so far is an individual list of reductions for that variable which would take precedence over the default reduction operations to perform.

- `IOBasic::outInfo_reductions` (steerable)
  The default list of global reduction operations to perform on CCTK_GF and CCTK_ARRAY variables. This setting can be overridden for individual variables using an option string.
  Multiple reduction names must be separated by spaces.

All of the above parameters marked as steerable can be changed at runtime.

3 Examples

3.1 Example for `Info` Output

The following parameter settings request info output for variables `grid::r, wavetoy::phi` (both are CCTK grid functions) and `mythorn::complex` (a complex CCTK scalar) at every other iteration.

The minimum and maximum of `grid::r` is printed according to the list of default reductions for info output (parameter `IOBasic::outInfo_reductions`). This list is overridden for `wavetoy::phi` where only the L2 norm is output as specified in the option string for this variable. You can also add other reduction operators within the {} braces.

For the scalar variable `mythorn::complex` both the real and imaginary part are printed.

```
IOBasic::outInfo_every      = 2
IOBasic::outInfo_vars       = "grid::r
                               wavetoy::phi{reductions = 'norm2'}
                               mythorn::complex"
IOBasic::outInfo_reductions = "minimum maximum"
```

The resulting screen output would look like this:

```
-------------------------------------------------------------------------------------------------
  it  |          | GRID::r                    | WAVETOY::phi | MYTHORN::complex            |
      |    t     | minimum      | maximum     | norm2        | real part    | imag part    |
-------------------------------------------------------------------------------------------------
    0 |    0.000 |   0.02986294 |   0.86602540 |   0.04217014 |   6.90359593 |   0.00000000 |
    2 |    0.034 |   0.02986294 |   0.86602540 |   0.00934749 |   6.90359593 |   0.00000000 |
    4 |    0.069 |   0.02986294 |   0.86602540 |   0.02989811 |   6.90359593 |   0.00000000 |
    6 |    0.103 |   0.02986294 |   0.86602540 |   0.05899959 |   6.90359593 |   0.00000000 |
    8 |    0.138 |   0.02986294 |   0.86602540 |   0.07351147 |   6.90359593 |   0.00000000 |
   10 |    0.172 |   0.02986294 |   0.86602540 |   0.07781795 |   6.90359593 |   0.00000000 |
```

3.2 Example for `Scalar` Output

The following parameter settings request scalar output for all grid function variables in the group `grid::coordinates` and for the scalar variable `grid::coarse_dx`.
Output occurs every 10th iteration. `gnuplot` output style is selected for the ASCII files which are placed into a subdirectory `scalar_output`. The minimum and maximum of `grid::r` is printed according to the list of default reductions for scalar output (parameter `IOBasic::outScalar_reductions`). This list is overridden for `wavetoy::phi` where only the L1 norm is output as specified in the option string for this variable. You can also add other reduction operators within the {} braces.

```
IOBasic::outScalar_every      = 10
IOBasic::outScalar_vars       = "grid::coordinates grid::coarse_dx wavetoy::phi{'norm1'}"
IOBasic::outScalar_reductions = "minimum maximum"
IOBasic::outScalar_style      = "gnuplot"
IOBasic::out_dir              = "scalar_output"
```

This would create the following ASCII files:

```
~/Cactus/par> ls scalar_output
coarse_dx.asc  r_minimum.asc  x_minimum.asc  y_minimum.asc  z_minimum.asc
r_maximum.asc  x_maximum.asc  y_maximum.asc  z_maximum.asc  phi_norm1.asc
```

4 Comments

Possible Reduction Operations
In order to get output of reduction values for `CCTK_GF` and `CCTK_ARRAY` variables you need to activate a

thorn which provides reduction operators (eg. thorn `PUGHReduce` in the `CactusPUGH` arrangement). For a list of possible reduction operations please refer to the documention of this reduction thorn.

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 of 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 **IOBasic** 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.