PUGHReduce

Gabrielle Allen, Thomas Radke

Date: 2003/04/15 15:31:12

Abstract

Reductions operations which are performed using the PUGH driver

1 Purpose

This thorn registers a number of reduction operators with the flesh. The reductions are performed using internals of the PUGH driver, so that this thorn can only be used when CactusPUGH/PUGH is active.

The reduction operations this thorn registers are

<table>
<thead>
<tr>
<th>Reduction Operator</th>
<th>Calculates</th>
<th>By</th>
</tr>
</thead>
<tbody>
<tr>
<td>average*, mean*</td>
<td>the average/mean of a grid variable</td>
<td>$\frac{\sum GV}{N}$</td>
</tr>
<tr>
<td>count</td>
<td>the number of grid points in a grid variable</td>
<td>$N$</td>
</tr>
<tr>
<td>maximum*</td>
<td>the maximum of a grid variable</td>
<td>$\text{max} GV$</td>
</tr>
<tr>
<td>minimum*</td>
<td>the minimum of a grid variable</td>
<td>$\text{min} GV$</td>
</tr>
<tr>
<td>norm1, L1Norm</td>
<td>the L1 norm of a grid variable</td>
<td>$\frac{\sum</td>
</tr>
<tr>
<td>norm2, L2Norm</td>
<td>the L2 norm of a grid variable</td>
<td>$\sqrt{\frac{\sum</td>
</tr>
<tr>
<td>norm3, L3Norm</td>
<td>the L3 norm of a grid variable</td>
<td>$\sqrt[3]{\frac{\sum</td>
</tr>
<tr>
<td>norm4, L4Norm</td>
<td>the L4 norm of a grid variable</td>
<td>$\sqrt[4]{\frac{\sum</td>
</tr>
<tr>
<td>norm_inf, LinfNorm</td>
<td>the Infinitity norm of a grid variable</td>
<td>$\max</td>
</tr>
<tr>
<td>sum*</td>
<td>the sum of the elements of a grid variable</td>
<td>$\sum GV$</td>
</tr>
</tbody>
</table>

Reduction operators with multiple names are just synonyms for the same kind of reduction operation. In the formulas $GV$ is the grid variable to be reduced, and $N$ denotes the number of its elements. Reduction operators marked with * cannot be applied to grid variables of complex datatype.

2 Examples

The following C example illustrates how the get the maximum value of a grid function.

```c
int vindex;    /* grid variable index */
CCTK_REAL result; /* resulting reduction value */
int target_proc; /* processor to hold the result */
int reduction_handle; /* handle for reduction operator */
char *reduction_name; /* reduction operator to use */

/* want to get the maximum for the wavetoy grid function */
reduction_name = "maximum";
vindex = CCTK_VarIndex ("wavetoy::phi");

/* the reduction result will be obtained by processor 0 only */
target_proc = 0;
```
/* get the handle for the given reduction operator */
reduction_handle = CCTK_ReductionHandle (reduction_name);
if (reduction_handle >= 0)
{
    /* now do the reduction using the flesh’s generic reduction API 
       (passing in one input, expecting one output value of REAL type) */
    if (CCTK_Reduce (cctkGH, target_proc, reduction_handle,
                    1, CCTK_VARIABLE_REAL, &result, 1, vindex) == 0)
    {
        if (CCTK_MyProc (cctkGH) == target_proc)
        {
            printf("%s reduction value is %f\n", reduction_name, result);
        }
    }
    else
    {
        CCTK_VWarn (1, __LINE__, __FILE__, CCTK_THORNSTRING,
                     "%s reduction failed", reduction_name);
    }
    else
    {
        CCTK_VWarn (1, __LINE__, __FILE__, CCTK_THORNSTRING,
                     "Invalid reduction operator '%s'", reduction_name);
    }
}