

Thorn Guide for the **LocalInterp** Thorn

Thomas Radke

Date: 2004/06/27 11:00:14

Abstract

This thorn does processor-local interpolation of N-dimensional data arrays. In general there may be many input arrays (all defined on the same uniform Cartesian grid) all being interpolated to the same set of interpolation points.

1 Introduction

This thorn provides processor-local interpolation, using the interpolation operator

"uniform cartesian"

for the Cactus local interpolation API `CCTK_InterpLocalUniform()`. (Note that the word "cartesian" is in lower case here!) It supports 1, 2, and 3-dimensional interpolation.

See the Cactus Reference Manual for a full description of the `CCTK_InterpLocalUniform()` API, and some examples of how to use it.

1.1 History

This interpolator was written by Thomas Radke in early 2001 (drawing on older code by Paul Walker). It originally lived in the `PUGHInterp` thorn, but it turned to have very little to do with `PUGH`, so was moved here in winter 2001-2002.

From winter 2001-2002 to July 2003 this thorn also contained another interpolator written by Jonathan Thornburg, but in July 2003 that interpolator was moved to `AEIThorns/AEILocalInterp/` because it was (is) GPL and Cactus policies are that this arrangement (`CactusBase`) is reserved for code under the Cactus-flesh license (= GPL except that it's like LGPL for linking with other thorns).

Because CVS can't delete directories, this thorn still contains a lot of empty-except-for-CVS-directories directory trees left over from Jonathan Thornburg's interpolator. You can/should ignore these.

2 Implementation

Internally, this interpolator always does 3-D interpolation, inserting zero coordinates as appropriate for lower dimensionalities. The interpolation is done by successive 1-D interpolations along each axis.¹ See the `README` file in the `src/` directory for further details.

2.1 Additional information passed in as table options

The `CCTK_InterpLocalUniform()` API accepts a table handle as one of its arguments which can be used to pass additional information to the local interpolator via table options.

The only table option supported so far by `LocalInterp`'s "uniform cartesian" operator is the interpolation order which must be passed as a `CCTK_INT` value with key "order". Options with keys

¹Note that this means that different axes are treated slightly differently by the interpolator. In other words, at the level of finite differencing errors, interpolation does *not* commute with permuting the axes. However, in practice the differences are likely to be small, at least for smooth input data.

"N_boundary_points_to_omit", "boundary_off_centering_tolerance", or "boundary_extrapolation_tolerance" (which are usually passed by a global interpolator) are also recognized but silently ignored (a level-4 warning message will be issued in this case).