The Berger & Oliger Algorithm Using a Shadow Hierarchy for Truncation Error Estimation

Matthew W. Choptuik
Center for Relativity
The University of Texas at Austin
Austin, TX, 78712-1081
matt@infeld.ph.utexas.edu

May 1995

1 Overview

The basic idea of this implementation is to continuously integrate the equations of motion (difference equations) on a coarsened version (the shadow hierarchy) of the base Berger & Oliger grid hierarchy, as well as on the base hierarchy itself. Conceptually, this strategy significantly simplifies the truncation error estimation procedure compared, for example, to the procedure used in `ad`. More importantly obviates the need to replicate fine-grid storage at regridding times. These benefits come at the cost of a relatively small amount of additional computation and increased communication but the hope is that these will represent a small price to pay for the generality of this approach. It should be noted, however, that any restrictions on grid geometries (minimum number of grid points on an edge, blocking factors, etc.) should fundamentally be imposed on the *shadow hierarchy* rather than the base hierarchy.

Let G^l_j and $\tilde{G}^l_j$ denote the base and shadow hierarchies respectively. Here the superscript l's label the level in the hierarchy while the subscript j's label grids or bounding boxes at a particular level. Let L denote the maximum possible *total number* of levels of refinement to be used in the calculation. Then we can conveniently take $l = 1 \ldots L$ for G^l_j and $l = 0 \ldots L-1$ for $\tilde{G}^l_j$. At all times, the grid structure defined by $\tilde{G}^l_j$ is to be a precise 2:1 coarsening of that defined by G^l_j. This implies that the mesh-refinement factor, ρ, (assumed constant here) must satisfy $\rho = 2^p$, which, particularly for 3-d calculations, is unlikely to be an undue restriction. As stated above, the idea is to integrate equations on $\tilde{G}^l_j$ in synchrony with those on G^l_j. Thus, the entire Berger & Oliger update procedure is applied to both G^l_j and $\tilde{G}^l_j$. The only stage where treatment of the hierarchies differs is after regridding at level l, wherein the level $l-1$ unknowns on $\tilde{G}^{l-1}_j$ (which logically correspond to the level l unknowns on G^l_j) are reinitialized via restriction from the level l values on G^l_j.

2 Pseudo-code

The core of the algorithm is defined by the following pseudo-code:

routine `cycle`$[l, n]$
    # *l := level of base hierarchy being updated*
    # *n := time step on this level (relative to last parental step for $l > 1$, initial time for $l = 1$)*
    # *$l^\star$:= current finest level in G^l_j*
    # t_l := integration time (physical time) on level l, hierarchy G^l_j
    # $\tilde{t}_l$:= integration time (physical time) on level l, hierarchy $\tilde{G}^l_j$
    # ρ := mesh refinement ratio (nominally 2)
    if regridding time **and** $l < L$ **then**
        `regrid` on *all* existing levels $l' \geq l$
        `recompose` $G^{l+1 \cdots l^\star}_j$ (incl. transfer of old $l' \geq l$ values and prolongations from parents)
        `recompose` $\tilde{G}^{l \cdots l^\star - 1}_j$ and reinitialize $\tilde{G}^{l-1}_j$ via restriction from G^l_j
    end if
    # *Integrate Base Hierarchy*
    if $l > 1$ **then**
        `set_boundary_values` for G^l_j values from $G^{l-1}_{j'}$ values
    end if
    `update` difference equations on G^l_j
    if `mod`$(n, 2) = 0$ **then**
        # *Integrate Shadow Hierarchy*
        if $l > 1$ **then**
            `set_boundary_values` for $\tilde{G}^{l-1}_j$ values from $\tilde{G}^{l-2}_{j'}$ values
        end if
        `update` difference equations on $\tilde{G}^{l-1}_j$
    end if
    if $l < l^\star$ **then**
        for $n' = 0 \cdots \rho - 1$
            `cycle`$[l + 1, n']$
        end for
    end if
    if $l > 1$ **and** $t_l = t_{l-1}$ **then**
        `restrict` G^l_j values to G^{l-1}_j
        if $\tilde{t}_{l-1} = \tilde{t}_{l-2}$ **then**
            `restrict` $\tilde{G}^{l-1}_j$ values to $\tilde{G}^{l-2}_j$
        end if
    end if
end routine