CEMRACS 98

Object oriented conception of Adaptive Mesh Refinement (AMR) algorithm for structured meshes

Nicolas Huré (hure@onera.fr) - Christophe Prud'homme

The purpose of AMR algorithms is to concentrate the computational effort where necessary. AMR is based on a sequence of nested grids with finer and finer mesh. These fine grids are recursively embedded in coarser grids. The accuracy of the solution is automatically estimated and rectangular structured fine grid patches are dynamically created or removed to achieve a desired accuracy. Special difference equations are used at the interface between coarse and fine grids to enforce conservation.
The following operations are typical of the AMR algorithm:
- The hierarchy of grids must be properly nested [Quirk]
- The conservation laws are enforced for all coarse grid cells overlain by fine grid cells and at boundaries between fine and coarse grids.
- Boundary informations may be taken from adjoining grids of the same level l, adjoining grids of level $(l-1)$, or from the boundary condition.
- When a level l is computed, cells with high error are tagged and level $(l+1)$ created.

All these operations require manipulation of complex data structures. AMR is a dynamic and recursive algorithm. These concepts can not be expressed easily in Fortran.
Object oriented approach and C++ simplifies the implementation. C++ is recursive and it is easy to make the code evoluate and to change a method (solver, error estimation, ...).
A good way is to use the Standard Template Library (STL), who decreases the algorithm complexity, and Blitz++ who gives an optimized class array.
With Blitz++, algorithms uses Matlab like arrays and matrices expression.
With STL, we can chose the following data structure for AMR:
- A "patch" object contains coordinates, boundary conditions (in ghost cells), solution and is localized using the index of the previous level patch. The solver works on a patch without other infomations.
- A "level" is an ordered *set*<> of "patch" objects. With STL we define an order relation. Then a new created patch is automatically inserted at the right place.
- The AMR levels hierarchy is a simple *list*<> of "level" objects. Before a new adaptation, it is easy to delete all the hierarchy.

AMR is quite complex method, but fortunately using appropriate tools (STL, Blitz) and a high level language, the complexity can be reduced easily.