Introduction

In spectral methods refinement takes place by decreasing the size of the elements (h-refinement) or increasing the order of the approximation (N-refinement). Adaptive mesh refinement depends on the experienced selections of tolerances and refinement criteria that are highly problem-dependent. The goal of the current work is to implement a method of a high-order adaptive code based on a nonconforming spectral element method [1].

Non-conforming Mesh

There are a few non-conforming methods that place no restriction on the size, location, or connectivity of the elemental subdomains. These methods fall under the general classification of PATCHING. In this case we have to connect the solution on two subdomains separated by an interface called a PATCH, see Fig. 1.0.

![Patch](Image)

The PATCH is constructed from a set of line segments $\gamma_i$ associated with a particular set of elemental edges $I_i$. This defines a set of non-conforming interfaces, where the continuity cannot be guaranteed in the discrete solution.

Refinement criteria

The mesh refinement strategies are based on h2-refinement of quadrilaterals, i.e. a quadrilateral can be divided either vertically and horizontally. Here we consider two types of refinement criteria:

1. Refine everywhere that solution gradients are large, requiring
   \[ \| \nabla u^{(k)} \| \leq \epsilon \| u^h \|_1 \]
everywhere in the mesh, where \( \| \cdot \| \) is the $L_2$ norm, and \( \| \cdot \|_1 \) is the $H_1$ norm, and \( \epsilon \) is the discretization tolerance.

2. The second criteria takes direct advantage of the high-order polynomial basis. Consider the expansion of a given smooth function $u$ over the domain $D = [-1, 1]^2$ in terms of Legendre polynomials
   \[ u(x, y) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} a_{n,m} P_n(x) P_m(y) \]
   where the expansion coefficients are given by
   \[ a_{n,m} = \frac{1}{c_n c_m} \int_{-1}^1 \int_{-1}^1 P_n(x) P_m(y) \ | J \ | \ dx \ dy \]
   where the normalization constant is $c_i = (i + 1/2)^{-1}$. We can form an estimate of the approximation error $\| u - u^h \|$ by examining the tail of the spectrum:
   \[ a_p = | a_{p,p} | + \sum_{i=0}^{p-1} | a_{i,p} | + | a_{p,i} |. \]

For the accurate representation of $u$ we require the spectrum to satisfy the discretization tolerance:
   \[ | a_{p,p} | + \sum_{i=0}^{p-1} | a_{i,p} | + | a_{p,i} | \leq \epsilon \| u^h \| \]

Implementation

The algorithm for mesh refinement has been implemented using the C++ language and for the computational modules Fortran (SEPRAN). The most complex problem is maintaining the connectivity of the mesh dynamically. The conform mesh is produced by SEPRAN, and the C++ refine program uses this mesh as input.

Conclusions and Future Work

We have outlined the basis features of an adaptive spectral element method. The most interesting part of the method is the refinement criteria provided by the local polynomial spectrum. Also the method can be used to run the computations in parallel. To reduce the complexity of the stiffness matrices arising in spectral element methods, the static condensation algorithm will be used.

References: