Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

universitätbonn

Computer lab Numerical Algorithms
Winter term 2012/2013
Prof. Dr. M. Rumpf – B. Geihe, B. Heeren, S. Tölkes

Problem sheet 2 **November 6th, 2012**

In this exercise adaptive grid refinements based on local error indicators will be added to the cell problem considered on problem sheet 1. To assure that the refinement procedure takes account of the periodic boundary condition a new specialized grid class PeriodicSquare has already been added which should be used in this case. Your task is to complete the error estimator class DiffusiveErrorEstimator and use it in your main program to steer iterative adaptive refinements of the computational domain.
Upon construction the error estimator class receives references to the current grid, diffusivity tensor, right hand side, and numerical solution u_h, and an additional factor $0 < \gamma < 1$. Calling DiffusiveErrorEstimator :: markTriangles will create a Vector containing the local error indicators η_T for each element T of the triangulation. Then all elements whose error indicator exceeds $\gamma \max_T \eta_T$ will be marked for refinement. The actual refinement is then triggered by calling Grid :: refineMarkedTriangles. This method already accounts for the periodicity of the grid: When refining a triangle with one or more edges on the boundary of the domain, the periodic counterparts of these edges are identified and refined correctly.

Tasks:

- Implement an error indicator based on steep transitions in the diffusivity tensor:

  Algorithm 1: Computation of an error indicator η_T

  foreach *neighbour triangle T' of T* **do**
  $\eta_T \mathrel{+}= |T \cap T'| \frac{|a(T.center) - a(T'.center)|}{\|T.center() - T'.center()\|}$

  You can check a Triangle for neighbours by comparing Triangle :: getNeighbourIndex to IndexNotSet. The neighbouring triangle can be accessed via Triangle :: getNeighbour

- Write your main program by copying relevant parts from problem01 and do iterative refinements by using DiffusiveErrorEstimator.

Extra task: Implement a more sophisticated error estimator like e.g.

$$\eta_T = \|h(f + \mathrm{div}(a\nabla u_h))\|_{0,2,T}^2 + \sum_{E \in \mathcal{E}^0(T)} \|h^{\frac{1}{2}}[a\nabla u_h \cdot n_E]_E\|_{0,2,E}^2$$

Note: Contrary to the classes described above, the BlobTensor does not handle periodicity correctly. As this exercise is only an example intended for understanding error estimation and grid refinement we will ignore this fact for now. For the same reason it is acceptable to not handle the periodicity of the domain in error estimators *for now*. In real applications it is of course important to compute all factors correctly in order to achieve non-defective results.