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 4 **November 21st, 2012**

We now want to solve the macroscopic problem on a regular triangulation $\mathcal{T}_H = \mathcal{T}_H(D)$ of $D = [0,1]^2$: Find $u_H \in \mathring{\mathcal{V}}_H(D)$ such that

$$
\begin{aligned}
a_H(u_H, \varphi_H) &= (f, \varphi_H) \quad && \forall\, \varphi_H \in \mathring{\mathcal{V}}_H(D) \\
u_H &= 0 && \text{on } \partial D
\end{aligned}
$$

where $f \in L^2$ and

$$
a_H(\varphi_H, \psi_H) := \sum_{T \in \mathcal{T}_H} \sum_{\gamma_T \in T} \omega_{\gamma_T} \underbrace{ \fint_{\epsilon Q} a^{\epsilon}_{\gamma_T}(x)\, \nabla R_h(\varphi_H)(x) \cdot \nabla R_h(\psi_H)(x)\, \mathrm{d}x}_{=(a^{H,h}\, \nabla \varphi_H \cdot \nabla \psi_H)(\gamma_T)} .
$$

Here $R_h(\varphi_H)(x) \in \varphi_H(\gamma_T) + \nabla \varphi_H(\gamma_T)(x - \gamma_T) + \mathcal{V}_{h,\#}(\gamma_T)$ is given as the solution of the cell problem, i.e.

$$
\int_Q a^{\epsilon}_{\gamma_T}(x)\, \nabla R_h(\varphi_H)(x) \cdot \nabla \varphi_h(x)\, \mathrm{d}x = 0 \quad \forall\, \varphi_h \in \mathcal{V}_{h,\#}(\gamma_T) .
$$

The main task is now to implement the two-scale bilinear form $a^{H,h}$ in a class `TwoscaleBilf`. This class receives a bilinear form (e.g. `IsoDiffusiveStiffBilf`) as template argument and manages a list of micro cell problems. There is one micro cell for each quadrature point and in the most general case the quadrature rule used on the macro grid should be selectable. However, we assume that a one point quadrature rule is sufficient and hence there is currently only one micro cell problem per triangle.

To construct `TwoscaleBilf` a representative bilinear form a^{ϵ} is passed to the constructor. This bilinear form is then copied and stored internally. However, as the bilinear form on the mirco cell depends on the location of the quadrature point on the macro grid, the stored bilinear forms should be mutable. Hence one can manipulate them by means of `TwoscaleBilf::getBilfReference()`.

Whenever a bilinear form belonging to a micro cell has been manipulated, the class `TwoscaleBilf` has to be updated, i.e. the basis solutions of the affected micro cells have to be recomputed. To improve efficiency there is a member `TwoscaleBilf::cellProblemChanged` that knows which forms have been changed and should hence be updated.

Tasks:

- Complete the class TwoscaleBilf, i.e. fill TwoscaleBilf :: update() and TwoscaleBilf :: **operator**(). The latter one is supposed to construct base function gradients $\nabla \varphi_H^i$ and $\nabla \varphi_H^j$ and return $(a^{H,h} \nabla \varphi_H^i \cdot \nabla \varphi_H^j)(\gamma_T)$.
- Write a main program to test the two-scale bilinear form, e.g. compare the two-scale solution to a single-scale solution!