Lab 08 - Adaptive Refinement

Numerical Solution of PDEs Using the Finite Element Method

MHPC P2.13_seed

Martin Kronbichler kronbichler@lnm.mw.tum.de and **Luca Heltai** luca.heltai@sissa.it

---

1. The topic of this lab session is a modified version of step-4 made available for you and the goal is to implement adaptive refinement based on step-6, see https://www.dealii.org/8.5.0/doxygen/deal.II/step_6.html
2. Modify the included program to support adaptive refinement by adding a ConstraintMatrix and the necessary function calls in `setup` (to create the constraints), `assembly` (`distribute_local_to_global` and no `apply_boundary_values`), `solve` (`distribute`), and replace the logic in `refine_mesh`.
3. What happens if you "forget" the call `constraints.distribute(solution)` after solving the linear system?
4. Use `constraints.print (std::cout);` to show the constraints if you have the following mesh in 2d:

   ```
   GridGenerator::hyper_cube(triangulation);
   triangulation.refine_global (1);
   triangulation.begin_active()->set_refine_flag ();
   triangulation.execute_coarsening_and_refinement ();
   ```

   and also look at this mesh (for example in paraview). Does the result make sense to you? Also compare with/without adding boundary conditions!
5. Change the problem to be an L-shape in 3d (start with 2 global refinements) and manufactured solution

   $$u(x, y, z) = \frac{1}{r^2 + 0.001}.$$

   Make sure you apply the correct right-hand side and boundary conditions.
6. Compute the L2 error with respect to the number of DoFs with a) global refinement and

   b) adaptive refinement using a `KellyEstimator` like in step-6.
7. Plot the two graphs in 4) in a log-log plot and include the line for a quadratic rate. Note that $h^2 = N^{-2/3}$.