Lab 05

Residual-based error estimators.
deal.II step-6, step-14.

Advanced Topic in Scientific Computing - SISSA, UniTS, 2024-2025

Pasquale Claudio Africa

24 Oct 2024

Assignment

In this exercise, you will extend the current `step-6` tutorial by implementing a residual-based error estimator and compare it to the Kelly error estimator.

For each cell (K), compute the internal error contribution as:

$$\eta_K^2 = \int_K h_K^2 \left(f + \nabla \cdot (a \nabla u_h)\right)^2 \, \mathrm{d}x,$$

where u_h is the numerical solution, a is the diffusion coefficient, and f is the forcing term.

Use the same mesh refinement criteria and compare the refinement pattern and efficiency between the internal residual-based and Kelly error estimators, also by exporting the corresponding estimators to file.

Hint

Here is a template to compute the residual-based error estimator:

```cpp
template <int dim>
void Step6<dim>::compute_residual_based_error(Vector<float> &error_per_cell)
{
    const QGauss<dim> quadrature_formula(fe.degree + 1);
    FEValues<dim> fe_values(fe, quadrature_formula,
                            update_flags...);

    for (const auto &cell : dof_handler.active_cell_iterators())
    {
        fe_values.reinit(cell);
        double residual = 0.0;

        // Compute cell residuals (for internal elements).
        // Sum up the error contributions and store in error_per_cell[cell].

        error_per_cell[cell->active_cell_index()] = std::sqrt(residual);
    }
}
```

Bonus challenge

Following `step-14` (consider in particular the `integrate_over_regular_face` method), add face jump terms to the internal residual estimator:

$$\eta_K = h_K \left(\int_K (f + \nabla \cdot (a \nabla u_h))^2 \, \mathrm{d}x \right)^{\frac{1}{2}} + \frac{1}{2} h_K^{\frac{1}{2}} \left(\sum_{\text{faces}} \int_{\text{face}} ([a \nabla u_h \cdot n])^2 \, \mathrm{d}s \right)^{\frac{1}{2}},$$

and compare the results. As an alternative, properly combine the residual-based error estimator defined above with the output coming from the Kelly estimator.

For simplicity, assume that no hanging nodes are present.