

# Lab 05

---

**Residual-based error estimators.  
Convection-diffusion problems.  
deal.II step-6, step-9.**

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

Pasquale Claudio Africa

24 Oct 2024

# Assignment 1

---

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:

$$\eta_K^2 = \int_K h_K^2 (f + \nabla \cdot (a \nabla u_h))^2 dx,$$

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:

```
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:

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

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.

# Assignment 2

---

In this exercise, you will extend the problem in `step-6` by adding a convection term to the partial differential equation (PDE) being solved. Convection-dominated problems are often prone to numerical instabilities when solved using standard finite element methods without stabilization techniques.

# Step-by-step instructions (1/2)

---

## 1. Review the current formulation:

The current problem involves solving the following Laplace equation:

$$-\nabla \cdot (a(\mathbf{x})\nabla u) = f \quad \text{in } \Omega,$$

where  $a$  is the diffusion coefficient.

## 2. Modify the equation:

Add a convection term to the equation, resulting in:

$$-\nabla \cdot (a(\mathbf{x})\nabla u) + \mathbf{b} \cdot \nabla u = f \quad \text{in } \Omega,$$

where  $\mathbf{b}$  is the convection vector. The new term  $\mathbf{b} \cdot \nabla u$  models the convection.

## 3. Update the weak form:

Modify the weak form of the equation to account for the convection term. Implement this weak form in your finite element code by updating the relevant matrix assembly functions.

## Step-by-step instructions (2/2)


---

### 4. Set up the diffusion coefficient and the convection vector $\mathbf{b}$ :

For simplicity, you can use, for example:

$$a(\mathbf{x}) = 1, \quad \mathbf{b} = [1000, 1000]^T.$$

Update the source code to include this convection vector in the matrix assembly.

 **Warning:** with convection terms, the resulting matrix is not symmetric anymore! Choose proper linear solver and preconditioners.

### 5. Run the code and analyze the results:

Compile and run the modified problem. Visualize the numerical solution. In convection-dominated scenarios, you should observe oscillations or numerical instabilities in regions where convection dominates diffusion. This instability is due to the fact that the standard Galerkin method does not provide sufficient stability for convection-dominated problems.

# Questions to explore

---

- What happens to the solution as the convection term becomes stronger?
- How does the mesh refinement affect the solution stability?
- What types of numerical instabilities do you observe, and where do they occur in the domain?

## Bonus challenge

Once you observe the instabilities, try implementing a simple stabilization technique, such as artificial diffusion or SUPG, as discussed in `step-9`, to improve the numerical stability of the solution. Compare the results with and without stabilization.