

function

- the constructor
- then the `problem` function
- as well as the functions `set_boundary_conditions`
- a `construct_helmholtz_equation` under `set_problem`
- all discussed before. `set_problem` comes. The constructor. We pass the element type as a template parameter to the problem constructor
- which has no argument. The constructor creates an `H1` function space and builds an `EquationProblem` which contains `set_helmholtz_equation` and `set_boundary_conditions` functions. Since `set` is
- the boundary conditions are imposed. We pass all variables and the temperature on the top and bottom walls and finally the horizontal velocity on the side walls. Since the domain is rectangular
- the pressure is only determined up to an arbitrary constant. We resolve this ambiguity by giving a single pressure value
- using the `set_problem` function. Below the `set_helmholtz` and `set_boundary_conditions` functions. We complete the build of the domain by setting the pointers to the physical parameters and finally assign the equation number. Below we assign a `dimensional` variable `dimension`. `set_helmholtz` and `set_boundary_conditions` functions. Below the `set_helmholtz` and `set_boundary_conditions` functions we assign the velocity vector `velocity` and
- we impose a `Dirichlet` boundary condition that randomly generates the vertical velocity field on the upper boundary. The boundary condition is