

C++ code 3.8.20: Implementation class **EvalResponse**. → [GITHUB](#)

```
2  EvalResponse::EvalResponse(
3      const std::shared_ptr<lf::uscalfe::FeSpaceLagrangeO1<double>> &fe_space_p) {
4      // Basis vectors for 2D Euclidean space ("unit vectors")
5      Eigen::Vector2d e0{1.0, 0.0}, e1{0.0, 1.0};
6      // Compute approximate solutions
7      Eigen::VectorXd approx_sol_e0, approx_sol_e1;
8      approx_sol_e0 = solveImpedanceBVP(fe_space_p, e0);
9      approx_sol_e1 = solveImpedanceBVP(fe_space_p, e1);
10     // Initialize the
11     F_Mat_(0, 0) = computeBoundaryOutputFunctional(approx_sol_e0, fe_space_p, e0);
12     F_Mat_(0, 1) = computeBoundaryOutputFunctional(approx_sol_e1, fe_space_p, e0);
13     F_Mat_(1, 0) = computeBoundaryOutputFunctional(approx_sol_e0, fe_space_p, e1);
14     F_Mat_(1, 1) = computeBoundaryOutputFunctional(approx_sol_e1, fe_space_p, e1);
15 }
16
17 /* Operator overloading */
18 double EvalResponse::operator()(Eigen::Vector2d g, Eigen::Vector2d d) const {
19     double value;
20     value = d.dot(F_Mat_ * g);
21     return value;
22 }
```