

```

2  std::pair<Eigen::VectorXd, Eigen::VectorXd> solveTemperatureEvolution(
3      const If::assemble::DofHandler &dofh, unsigned int m, double cool_coeff,
4      Eigen::VectorXd initial_temperature_vec) {
5      std::pair<Eigen::VectorXd, Eigen::VectorXd> solution_pair;
6      double tau = 1.0 / m; // step size
7      const If::uscalfe::size_type N_dofs(dofh.NumDofs()); // dim. of FE space
8      std::cout << "\n>> SolveTemperatureEvolution: m = " << m
9                  << ", N = " << N_dofs << ", cool_coeff= " << cool_coeff << "\n"
10                 << std::endl;
11
12     // Initialization for the energies of the solution at times t = 0, tau, 2*tau,
13     // etc... This information is not required to solve the parabolic evolution
14     Eigen::VectorXd energies(m + 1);
15
16     // Precomputing the required data for SDIRK-2
17     SDIRK2TimeStepper SDIRK2_stepper(dofh, tau, cool_coeff);
18
19     std::cout << "\n>> Iterating the action of discreteEvolutionOperator"
20              << std::endl;
21     // Starting the evolution using the initial conditions
22     Eigen::VectorXd discrete_solution_cur =
23         SDIRK2_stepper.discreteEvolutionOperator(initial_temperature_vec);
24     energies[0] = thermalEnergy(dofh, initial_temperature_vec);
25     energies[1] = thermalEnergy(dofh, discrete_solution_cur);
26     // Evolving the parabolic temperature system
27     // While less elegant, we use a current and next step solution vector in
28     // the iteration to stay away from potential harming aliasing effects of
29     // putting an Eigen::Vector on both sides of an assignment statement.
30     Eigen::VectorXd discrete_solution_next;
31     for (int i = 1; i < m; i++) {
32         discrete_solution_next =
33             SDIRK2_stepper.discreteEvolutionOperator(discrete_solution_cur);
34         discrete_solution_cur = discrete_solution_next;
35         energies[i + 1] = thermalEnergy(dofh, discrete_solution_cur);
36     }
37     solution_pair = std::make_pair(discrete_solution_cur, energies);
38     return solution_pair;
39 } // solveTemperatureEvolution

```