

C++11 code 9.1.11: Implementation of `twoStageRadauTimeSteppingLinScalODE()`

→ [GitLab](#)

```
2  std::vector<double> twoStageRadauTimeSteppingLinScalODE(unsigned int m) {
3      std::vector<double> sol_vec;
4      double step_size = 5.0 / m; // Timestep "tau"
5      sol_vec.push_back(1.0);      // Initial value
6
7      // Discrete evolution operator: For the two-stage Radau method applied to the
8      // scalar linear ODE  $(d/dt)y = -y$ , this turns out to be a scalar valued
9      // function depending on the step_size. Since we take equidistant step sizes
10     // in this example, the action of the evolution operator is simply
11     // multiplication by a constant double
12     double evolution_op =
13         (1 - (step_size * (1 + step_size / 6.0)) /
14          ((1 + step_size * 5.0 / 12.0) * (1 + step_size / 4.0) +
15           step_size * step_size / 16.0));
16     // Compute discrete evolution by applying the evolution operator at each step
17     for (int i = 1; i < m + 1; i++) {
18         sol_vec.push_back(evolution_op * sol_vec.at(i - 1));
19     }
20     return sol_vec;
21 }
```