

C++11 code 9.1.12: Implementation of `testConvergenceTwoStageRadauLinScalODE`

→ [GitLab](#)

```
2 void testConvergenceTwoStageRadauLinScalODE() {
3     constexpr int nlter = 10;           // total number of iterations
4     double max_norm_errors[nlter];      // errors vector for all approx. sols
5     double rates[nlter - 1];            // The rates of convergence
6     double avg_rate = 0.0;              // The average rate of convergence over all iterations
7
8     // Error between the approx solutions as given by the two stage Radau method
9     // and the exact solution vector computed from the analytic formula vector
10    // computed from the analytic formula
11    double diff; // temporary variable used to compute error at various nodes
12    std::vector<double> approx_sol_vec;
13    for (int k = 0; k < nlter; k++) {
14        unsigned int m = 10 * std::pow(2, k); // number of equidistant steps
15        double step_size = 5.0 / m;           // time step 'tau'
16        // Creating exact solution vector. This vector is created by evaluating the
17        // exact solution using the analytic formula  $y(t) = \exp(-t)$  at the
18        // equidistant nodes of the time steps.
19        std::vector<double> exact_sol_vec;
20        for (int i = 0; i < m + 1; i++) {
21            exact_sol_vec.push_back(std::exp(-i * step_size));
22        }
23        // Computing approximate solution
24        approx_sol_vec = twoStageRadauTimesteppingLinScalODE(m);
25        // Computing the error in the maximum norm
26        max_norm_errors[k] = 0;
27        for (int i = 0; i < m + 1; i++) {
28            diff = std::abs(approx_sol_vec.at(i) - exact_sol_vec.at(i));
29            max_norm_errors[k] =
30                (diff > max_norm_errors[k]) ? diff : max_norm_errors[k];
31        }
32    }
33    // Computing rates of convergence
34    for (int k = 0; k < nlter - 1; k++) {
35        rates[k] = log2(max_norm_errors[k] / max_norm_errors[k + 1]);
36        avg_rate += rates[k];
37    }
38    avg_rate = avg_rate / (nlter - 1);
```