

C++ code 5.2.55: Code for `brachistochrone_cvg()` → [GitLab](#)

```
2 void brachistochrone_cvg(std::string filename) {
3     // Set minimal number of cells in the mesh
4     int M = 8;
5     // Start on the cycloid curve (xi = 0.5)
6     Eigen::Vector2d a({0.5 * M_PI - 1.0, -1.0});
7     // Set b to the endpoint of the cycloid curve
8     Eigen::Vector2d b({M_PI, -2.});
9     // Initialize parameters for brachistochrone function
10    double atol = 1e-6;
11    double rtol = 1e-4;
12    double itmax = 100;
13    std::vector<double> L2errors;
14    std::ofstream outfile;
15    if (!filename.empty()) {
16        outfile.open(filename);
17    }
18
19    // Refine the mesh 8 times and compute the associated L2 errors
20    for (int j = 0; j < 8; ++j) {
21        // Compute the exact solution (i.e. the cycloid)
22        Eigen::Matrix<double, 2, Eigen::Dynamic> mu_cycloid(2, M + 1);
23        for (int i = 0; i <= M; ++i) {
24            const double xi = 0.5 * (static_cast<double>(i) / M + 1.0);
25            mu_cycloid(0, i) = M_PI * xi - std::sin(M_PI * xi);
26            mu_cycloid(1, i) = std::cos(M_PI * xi) - 1.;
27        }
28        // Solve the brachistochrone problem
29        Eigen::Matrix<double, 2, Eigen::Dynamic> mu =
30            brachistochrone(M, a, b, atol, rtol, itmax);
31        // Compute and store the L2 error
32        L2errors.push_back(L2norm(mu - mu_cycloid));
33        // Half the mesh width/double the number of cells
34        M *= 2;
35        if (!filename.empty() && outfile.good()) {
36            outfile << "mu{" << j + 1 << "} = [ ";
37            for (int k = 0; k < mu.cols(); ++k) {
38                outfile << mu(0, k) << " ";
39            }
40            outfile << ";" << std::endl;
41            for (int k = 0; k < mu.cols(); ++k) {
42                outfile << mu(1, k) << " ";
43            }
44            outfile << " ];" << std::endl;
45        }
46    }
47
48    // Print a table with results.
49    std::cout << "| k | ||u_h-I_M u_h||_{L2} |\n";
50    std::cout << "-----|\n";
51    for (int i = 0; i < L2errors.size() - 1; ++i) {
52        std::cout << "| " << std::setw(3) << 8 * std::pow(2, i) << " | "
53            << std::setw(20) << std::setprecision(5) << std::scientific
54            << L2errors.at(i) << " |\n";
55    }
56 }
```