

## C++ code 6.7.9: Sub-problem (6-7.e): Implementation of constructor of evaluation operator

```
2 Eigen::VectorXd CubicHermiteInterpolant::eval(const Eigen::VectorXd &x) const {
3     Eigen::VectorXd ret(x.size());
4
5     // TODO: (6-7.e) Implement evaluation function.
6     // START
7     unsigned int i_star = 0;
8     double tmp, h_tmp, t1, t2, y1, y2, c1, c2, a1, a2, a3;
9     for (unsigned int j = 0; j < x.size(); j) {
10         // Stores the current interval index and some temporary variable
11         // Find the interval for x_j while assuming that x values are sorted
12         // and within  $[t_{\min}, t_{\max}]$ 
13         if (t_(i_star) <= x(j) && x(j) <= t_(i_star + 1)) {
14             t1 = t_(i_star);
15             t2 = t_(i_star + 1);
16             h_tmp = t2 - t1; // h_tmp = h_ for equidistant grid
17             y1 = y_(i_star);
18             y2 = y_(i_star + 1);
19             c1 = c_(i_star);
20             c2 = c_(i_star + 1);
21             a1 = y2 - y1;
22             a2 = a1 - h_tmp * c1;
23             a3 = h_tmp * c2 - a1 - a2;
24
25             // Compute  $s(x(j))$ 
26             tmp = (x(j) - t1) / h_tmp;
27             ret(j) = y1 + (a1 + (a2 + a3 * tmp) * (tmp - 1.)) * tmp;
28
29             ++j;
30         } else {
31             // Otherwise, go to next interval
32             ++i_star;
33             // Terminate if outside any interval
34             if (i_star >= t_.size() - 1) break;
35             continue;
36         }
37     }
38     // END
39
40     return ret;
41 }
```

Get it on 🐘 GitLab (pchi.hpp).