

C++ code 6.7.7: Sub-problem (6-7.e): Definition of Hermite interpolant-class

```
2 class CubicHermiteInterpolant {
3 public:
4     /*!
5     * \brief Construct the slopes from the data.
6     * Use finite-differences or setting  $s'(x_j) = 0$ .
7     * \param[in] t Vector of nodes (assumed equidistant and sorted).
8     * \param[in] y Vector of values at nodes  $t$ .
9     * \param[in] s Flag to set if you want to reconstruct or set the slopes to
10    *zero.
11    */
12    template <typename Function>
13    CubicHermiteInterpolant(Function &&f, const Eigen::VectorXd &t);
14    virtual ~CubicHermiteInterpolant() = default;
15
16    /*!
17    * \brief Evaluate the interpolant at the nodes  $x$ .
18    * The input is assumed sorted, unique and inside the interval
19    * \param[in] x The vector of points  $x_i$  where to compute  $s(x_i)$ .
20    * \return Values of interpolant at  $x$  (vector)
21    */
22    Eigen::VectorXd eval(const Eigen::VectorXd &x) const;
23
24 private:
25    // Provided nodes and values  $(t, y)$  to compute spline,
26    // Eigen vectors,  $c$  contains slopes
27    // All have the same size  $n$ 
28    Eigen::VectorXd t_, y_, c_;
29    // Difference  $t(i) - t(i-1)$ 
30    double h_;
31    // Size of  $t$ ,  $y$  and  $c$ .
32    unsigned int n_;
33};
```

Get it on  GitLab (pchi.hpp).