provided that the functions $g$ are such that

$$g'(x_1) = f'(x_1) \quad \text{and} \quad g'(x_n) = f'(x_n).$$

The following theorem gives information about the approximating properties of cubic splines.

**Theorem 5.11.7** Let $s$ be a cubic spline that interpolates a four times continuously differentiable function $f$ at the points

$$a = x_1 < x_2 < \ldots < x_n = b,$$

with correct boundary conditions. Then

$$\max_{a \le x \le b} |s^{(r)}(x) - f^{(r)}(x)| < K_r(\beta) M h^{4-r}, \qquad r = 0, 1, 2, 3,$$

where $\beta = \max_{i,j} \left( \frac{h_i}{h_j} \right)$, $h = \max_i h_i$, $|f^{(4)}(x)| \le M$, $a \le x \le b$, and

$$K_0(\beta) = \tfrac{5}{384}, \qquad K_1(\beta) = \tfrac{1}{216}(9 + \sqrt{3}),$$
$$K_2(\beta) = \tfrac{1}{12}(1 + 3\beta), \qquad K_3(\beta) = \tfrac{1}{2}(1 + \beta^2).$$

The theorem shows that the interpolating spline and its derivatives, up to the third, converge uniformly to $f$ and its derivatives as the largest step size $h$ goes to zero. For equidistant knots, we have $\beta = 1$, and, for non-equidistant knots, $\beta$ increases when the ratio between the largest and smallest subinterval grows.

For interpolation with splines with correct boundary conditions, we have, in particular, $|s(x) - f(x)| = O(h^4)$; for interpolation with natural splines, on the other hand, we have $|s(x) - f(x)| = O(h^2)$ close to $x_1$ and $x_n$.

## Exercises

1. We want to compute $f(a) = \sqrt{a}$ for $1 < a < 4$, and we have very high requirements concerning speed.



a) One possible method is to interpolate linearly in an equidistant table. Which table size is needed if we require that $R_{\mathrm{XF}} + R_{\mathrm{T}}$ shall be smaller than $2\mu$? The computer is using the floating point system $(2, 23, -126, 127)$.

b) Another method is to perform one iteration with Newton–Raphson's method applied to the equation $f(x) = x^2 - a = 0$. The starting approximation is taken from a table (see Section 4.6). Which table size is needed if we require that the error after one iteration is smaller than $2\mu$?

c) The computational work is approximately the same in a) and b). Which method required the smallest table?

2. The following tables with correctly rounded values are given

| $x$ | $\sin x$ | $\cos x$ | $\cot x$ |
|---|---|---|---|
| 0.001 | 0.001000 | 1.000000 | 1000.0 |
| 0.002 | 0.002000 | 0.999998 | 499.999 |
| 0.003 | 0.003000 | 0.999996 | 333.332 |
| 0.004 | 0.004000 | 0.999992 | 249.999 |
| 0.005 | 0.005000 | 0.999988 | 199.998 |

Compute $\cot(0.0015)$ as accurately as possible:

a) by interpolation in the table for $\cot x$.
b) by interpolation in the tables for $\sin x$ and $\cos x$.
c) Estimate the error in b).
d) Explain the difference between the results in a) and b).

The argument 0.0015 is assumed to be exact.

3. Show that the value of the $k$th divided difference $f[x_1, x_2, \ldots, x_{k+1}]$ is independent of the order of the points $x_1, x_2, \ldots, x_{k+1}$.

4. Assume that a function is tabulated in the points

$$x_i = x_1 + (i-1)h, \qquad i = 1, 2, \ldots, 6.$$

The values are correctly rounded to $d$ decimals. Give an upper bound for the maximal error in $\Delta^5 f(x_1)$.

5. Derive a method for estimating $\int_a^b f(x)\,dx$ by interpolating $f$ by a linear spline with the knots $x_i = a + (i-1)(b-a)/(n-1)$, $i = 1, 2, \ldots, n$.

6. Show that the interpolating linear spline $s$ with knots $x_1, x_2, \ldots, x_n$ is the function that minimizes

$$\int_{x_1}^{x_n} (g'(x))^2\,dx,$$

among all functions $g$ such that $g(x_i) = f_i$, $i = 1, 2, \ldots, n$, and such that $\int_{x_1}^{x_n} (g'(x))^2\, dx$ is bounded.

7. Compute an approximation of $f(2.5)$ by interpolating the following data using a natural cubic spline:

| $x$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $f(x)$ | 0 | 3 | 4 | 4 |

8. a) Determine the natural cubic spline that interpolates the values

| $x$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $f(x)$ | 1 | 0 | 0.5 | 1 |

b) Compute $s(0.5)$, $s(1.5)$ and $s(2.5)$, and sketch $s(x)$. Alternatively, use a graphics program to plot $s(x)$.
c) Compute $s'(3)$.

## References

Much of the work in this area was made by Newton, as is also indicated in the names of the interpolation formulas. See, e.g.,

H. H. Goldstine, *A History of Numerical Analysis from the 16th through the 19th Century*, Springer Verlag, 1977.

There is an extensive classical theory, where different representations of interpolating polynomials are derived using operator calculus, see, e.g.,

C.-E. Fröberg, *Numerical Mathematics, Theory and Computer Applications*, The Benjamin/Cummings Publishing Company, Menlo Park, 1985.

A practically oriented presentation of splines is given in

C. deBoor, *A Practical Guide to Splines*, Springer Verlag, New York, 1978.

The theoretical aspects of splines are thoroughly discussed in

L. L. Schumaker, *Spline Functions: Basic Theory*, New York Wiley Corp., New York, 1981.

# 6 Differentiation and Richardson Extrapolation

## 6.1 Introduction

We want to compute numerically derivatives of a function that is only known at certain discrete points. To do this we may, e.g., determine an interpolating spline and differentiate that. This method is to be preferred when derivatives are to be computed for many values of the argument, or when the points where the function is known are not equidistant.

In the rest of this chapter, however, we assume that the function values are known at equidistant points, and that the approximate derivative values are needed only at a small number of points. In this case, it is simpler to interpolate the function by a polynomial of low degree, and differentiate this polynomial. We shall see that this is equivalent to approximating derivatives by difference quotients. As a matter of fact, we have already used this technique. In Chapter 5, we estimated the truncation error with the first neglected term, i.e., the derivative $f^{(k)} = d^k f/dx^k$ was estimated by the difference $\Delta^k f/h^k$ (cf. Theorem 5.4.7 and Formula (5.7.1)).

When we have derived different difference approximations to derivatives, we shall use such an approximation to illustrate Richardson extrapolation. With Richardson extrapolation, we can reduce the truncation error in the difference approximation. Richardson extrapolation is a very powerful technique, which we shall later use also in connection with numerical methods for integration and the numerical solution of ordinary differential equations.

Before continuing, we would like to recall the "big O" concept, which will be used in the following chapters. The notation $f(h) = O(h^p)$ as $h \to 0$