Note on a comparison of evaluation schemes for the interpolating polynomial

By Lonny B. Winrich*

In this note the computational efficiency of four methods for evaluating the interpolating polynomial is examined. The methods considered are the Lagrange representation, the Barycentric formula, Aitken's algorithm and Neville's algorithm. In general, the Barycentric formula is found to be best if the degree of the polynomial is large; for polynomials of low degree the Lagrange formula should be used when a large number of evaluations are required but Aitken's or Neville's algorithm is more efficient if few evaluations are needed.

Elementary textbooks in numerical analysis are often replete with different representations for the interpolating polynomial. Each such representation amounts to an algorithm for computing the value of the interpolating polynomial at some desired point. While some of these texts recommend certain schemes by default (i.e., by presenting only certain schemes) few if any authors give a comparison of them as computational algorithms. Such a comparison is not difficult, and one is presented here.

In this note I compare, on the basis of operations required, four evaluation schemes for the interpolating polynomial. I assume that data is given at the $n + 1$ points $x_0, x_1, \ldots, x_n$ and that we wish to interpolate a function whose values at these points are $y_0, y_1, \ldots, y_n$ respectively. This interpolation will be accomplished by a polynomial of degree n.

I consider the following schemes for calculation of the polynomial.

I. The Lagrange Representation (Hamming, 1962, p. 94)

$$P(x) = \sum_{k=0}^{n} y_k L_k(x),$$

$$L_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \left(\frac{x - x_i}{x_k - x_i}\right).$$

II. The Barycentric Formula (Hamming, 1962, p. 95)

$$P(x) = \frac{\sum_{k=0}^{n} \frac{A_k y_k}{x - x_k}}{\sum_{k=0}^{n} \frac{A_k}{x - x_k}},$$

$$A_k = \frac{1}{\prod_{\substack{i=0 \\ i \neq k}}^{n} (x_k - x_i)}.$$

III. Aitken's Algorithm (Henrici, 1964, p. 206)

$$P(x) = P_{n,n},$$

$$P_{k,0} = y_k, \quad k = 0, 1, \ldots, n,$$

$$P_{k,d+1} = \frac{(x_k - x)P_{d,d} - (x_d - x)P_{k,d}}{x_k - x_d},$$

$$k = d + 1, d + 2, \ldots, n.$$

Table 1

	LAGRANGE	BARYCENTRIC	AITKEN/NEVILLE
Setup Costs	$n(n+1)$ subtractions $n^2 - 1$ multiplications $n + 1$ divisions	$n(n+1)$ subtractions $n^2 - 1$ multiplications $n + 1$ divisions	$n(n+1)$ subtractions
Evaluation Costs	$n + 1$ subtractions n additions $n(n+1)$ multiplications	$n + 1$ subtractions $2n$ additions $n + 1$ multiplications $n + 2$ divisions	$\frac{n+2}{2}(n+1)$ subtractions $n(n+1)$ multiplications $\frac{n}{2}(n+1)$ divisions

* *University of Missouri-Rolla, Rolla, Missouri*

Downloaded from https://academic.oup.com/comjnl/article/12/2/154/356829 by guest on 21 August 2022

IV. Neville's Algorithm (Henrici, 1964, p. 208)

$$P(x) = P_{n,\,n},$$
$$P_{k,\,0} = y_k,\ k = 0, 1, \ldots, n,$$
$$P_{k,\,d+1} = \frac{(x_k - x)P_{k-1,\,d} - (x_{k-d-1} - x)P_{k,\,d}}{x_k - x_{k-d-1}}$$
$$k = d + 1, d + 2, \ldots, n.$$

Each of these algorithms requires certain coefficients which depend only on the data and can be calculated once. The cost of such calculation may be called the setup cost.

In the Lagrange scheme we set up by calculating

$$\frac{y_k}{\prod\limits_{\substack{i=0\\ i \neq k}}^{n} (x_k - x_i)},\ k = 0, 1, \ldots, n.$$

For the Barycentric formula the coefficients $A_k, k = 0, 1, \ldots, n$ need to be calculated just once. Aitken's and Neville's algorithms, require only the differences $x_k - x_d, k, d = 0, 1, 2, \ldots, n, k \neq d$.

In addition to these setup costs, we must also consider the cost of evaluating the polynomial for a particular value of x. The summary of these costs is given in **Table 1**. There is no cost difference between Aitken's and Neville's algorithms.

Some general observations can be made on the basis of this table. The setup costs of Aitken/Neville are clearly less than those of the other two methods; but the evaluation costs, because of the number of multiplications and divisions, are almost surely greater. Thus a trade-off in terms of number of evaluations is indicated. Furthermore, while the setup costs of the Lagrange scheme and the Barycentric formula are the same, the number of multiplications and divisions is a linear function of n for the Barycentric scheme but a quadratic function for the Lagrange method. Thus a trade-off based on the degree of the interpolating polynomial is indicated.

In order to define these trade-offs accurately, we must consider the relative times of the operations for a particular machine.

A typical set of ratios of operation times are:

Add or subtract = 1 time unit
Multiply = 2 time units
Divide = 3 time units

Using these ratios I indicate, in **Table 2,** the method requiring the least time by an L for Lagrange, B for Barycentric, or A/N for either Aitken's or Neville's algorithm.

Table 2

n = degree of interpolating polynomial
k = number of evaluations

<table>
<tr><th>n \ k</th><th>1</th><th>2, 3, . . .</th></tr>
<tr><td>1</td><td rowspan="3">A/N</td><td rowspan="3">L</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4
5
.
.
.</td><td colspan="2">B</td></tr>
</table>

On the IBM 360/50 at the University of Missouri-Rolla the approximate ratios for normalised, floating point, single precision, register-memory operations are:

Add or Subtract = 1 time unit
Multiply = 3·1 time units
Divide = 3·2 time units

Using these time ratios, we obtain the results shown in **Table 3**.

Table 3

<table>
<tr><th>n \ k</th><th>1</th><th>2</th><th>3, 4, . . .</th></tr>
<tr><td>1</td><td>A/N</td><td>A/N or L</td><td rowspan="2">L</td></tr>
<tr><td>2</td><td>A/N</td><td>L</td></tr>
<tr><td>3</td><td>A/N</td><td colspan="2" rowspan="2"></td></tr>
<tr><td>4</td><td>A/N</td></tr>
<tr><td>5
6
.
.
.</td><td colspan="3">B</td></tr>
</table>

Finally, we note that these results are modified slightly by changing the setup calculations. For example, we might calculate both A_k and $A_k y_k$ in setting up the Barycentric method or we might calculate the quotients $1/(x_k - x_d)$ for the Aitken/Neville scheme. These changes would, in general, increase the setup costs while decreasing the evaluation costs. The precise trade-offs for a given computer would, of course, change, but the general picture remains the same.

References

Hamming, R. W. (1962). *Numerical Methods for Scientists and Engineers*, McGraw-Hill, New York.
Henrici, Peter (1964). *Elements of Numerical Analysis,* John Wiley, New York.

Note added at the suggestion of the referee

Much better algorithms can be devised, and the general ones streamlined when tabular values are available *at equal intervals*, mainly by combining n divisions into a single final division, and by replacing certain multiplications by subtractions.

Downloaded from https://academic.oup.com/comjnl/article/12/2/154/356829 by guest on 21 August 2022