Journal of Computational and Applied Mathematics 16 (1986) 237–240
North-Holland

Comparison of spline and Lagrangian interpolation

John Michael McNAMEE
Computer Science Department, Atkinson College, York University, Downsview, Ontario M3J 2R7, Canada

Received 5 May 1985

Abstract: Errors in spline and Lagrangian methods of interpolation are compared for a range of functions, tabular intervals, and levels of data errors. For large step-sizes (an important case in practise) and fairly accurate data Lagrangian methods are considerably more accurate than splines.

1. Introduction

In recent years many authors of books on numerical methods have expressed the view that spline functions are superior to Lagrangian polynomial fits in respect to interpolation.

For example, Forsythe et al. [1, p. 70] refer to the concept of fitting data with a moving polynomial, e.g. passing a 10th degree polynomial through 11 points and using its values only for the central part of this interval. They then say: "Since spline interpolation works so much better, we shall not discuss moving polynomials further".

However, this author has never encountered proof that spline functions give more accurate approximations to function values than Lagrange type interpolation.

Indeed, numerical experiments to be described below seem to indicate that Lagrangian interpolation is much more accurate than cubic spline fitting in some cases, and never apprecia-bly worse.

2. The experiments

Seven different functions were tabulated at various step-sizes and for accuracies ranging from two significant figures up to seven (where the function value was less than one we rounded to d decimal places instead of *s* significant figures). The step-sizes ranged from a basic step, usually 0.1, up to 10 times that value.

A list of the functions and step-sizes used are shown in Table 1. In each case 41 points were tabulated. For each case, i.e. each combination of step-size and accuracy, Lagrangian interpolation was performed using the moving polynomial concept referred to above. For example, if a 4th degree polynomial was fitted through 5 tabular points, the polynomial was evaluated at points $\frac{1}{4}$-way, $\frac{1}{2}$-way and $\frac{3}{4}$-way between the 3rd and 4th tabular points fitted, thus

points evaluated
↓ ↓ ↓
1 2 3 4 5

Table 1
Functions and step-sizes used

No. function	Initial point of table	Step-sizes used
1. $\sin(x)$	0.0	0.1 (0.1) 1.0
2. $\exp(x)$	0.0	0.1 (0.1) 1.0
3. $\exp(x^2)$	0.0	0.1 (0.1) 0.4
4. $\log_3(x)$	0.01	0.01 (0.01) 0.1
	2.7	0.1 (0.1) 1.0
5. x^6	0.0	0.1 (0.1) 1.0
6. $1/x$	0.01	0.01 (0.01) 0.1
	0.01	0.1 (0.1) 1.0
7. $1/(1+25x^2)$	$-2.0, -3.0$, etc. according to step size	0.1 (0.1) 1.0

For a given set of 3 points of evaluation, interpolation was performed in this way using polynomials of degree 2 up to 28 in steps of 2; at the same time the error in the interpolated values was computed and the minimum error, as the degree varied, was recorded.

This process was repeated for as many different sets of 3 points as could be accomodated by the 41-entry table, i.e. for 41 points in all. The average of the 41 different minimum errors was computed and printed out.

An identical experiment was performed using cubic spline interpolation, i.e. fitting a series of cubic splines through 3, 5, 7, etc. up to 29 points.

The particular implementation of Lagrange interpolation used was 'Neville's Modified Method', described by MacLeod [3]. This was chosen because MacLeaod showed that it was the most stable of several implementations against rounding error.

Cubic spline interpolation was performed using the subroutines ICSCCU and ICSEVU, which form part of the I.M.S.L. package [2].

3. Results

The results of the above experiments, for selected accuracies and step-sizes and both interpolation methods are shown in Table 2. Results for imtermediate values were intermediate.

It is seen that for small step-sizes, or low accuracy, the two methods are about equally good. But for moderate to large step-sizes and fairly accurate to accurate data, the Lagrange method is much better, by a factor of up to 10000 in some cases.

This is important, for the case of large step-size would appear to be fairly common in Engineering applications; for example it may be expensive to evaluate a function at many points, or desirable to minimize the size of tables stored in small computers.

4. Explanation

Since cubic spline interpolation has a truncation error of $O(h^4)$ (see Shampine and Allen [4, p. 62], and Lagrangian interpolation has error up to $O(h^{28})$ (depending on degree used), it is to be expected that for moderately large h and fairly high accuracy Lagrange would do better.

Table 2
Comparison of errors in Langrange and spline interpolation

Function	Tabular interval	Accuracy	Errors	
			Lagrange	Spline
$\sin(x)$	0.1	2 dec. pl.	0.25 E-2	0.25 E-2
	0.1	7 dec. pl.	0.82 E-7	0.28 E-7
	0.3	2 dec. pl.	0.48 E-2	0.48 E-2
	0.3	7 dec. pl.	0.35 E-7	0.49 E-5
	0.7	2 dec. pl.	0.43 E-2	0.44 E-2
	0.7	7 dec. pl.	0.32 E-7	0.18 E-3
	1.0	2 dec. pl.	0.46 E-2	0.53 E-2
	1.0	5 dec. pl.	0.50 E-5	0.98 E-3
	1.0	7 dec. pl.	0.54 E-7	0.98 E-3
$\exp(x)$	0.1	2 sig. figs.	0.56 E-2	0.58 E-2
	0.1	7 sig. figs.	0.55 E-7	0.50 E-7
	1.0	2 sig. figs.	0.37 E-2	0.32 E-2
	1.0	5 sig. figs.	0.34 E-5	0.26 E-3
	1.0	7 sig. figs.	0.18 E-6	0.26 E-3
$\exp(x^2)$	0.1	2 sig. figs.	0.60 E-2	0.67 E-2
	0.1	7 sig. figs.	0.89 E-7	0.41 E-4
$\log_e(x)$	0.1	2 sig. figs.	0.13 E-1	0.14 E-1
	0.1	7 sig. figs.	0.14 E-6	0.15 E-6
	1.0	2 sig. figs.	0.43 E-2	0.41 E-2
	1.0	7 sig. figs.	0.50 E-7	0.62 E-7
x^6	0.1	2 sig. figs.	0.71 E-2	0.69 E-2
	0.1	7 sig. figs.	0.13 E-6	0.18 E-5
	1.0	2 sig. figs.	0.71 E-2	0.69 E-2
	1.0	7 sig. figs.	0.16 E-6	0.18 E-5
$1/x$	0.1	2 sig. figs.	0.21 E-2	0.21 E-2
	0.1	7 sig. figs.	0.23 E-7	0.13 E-6
	1.0	2 sig. figs.	0.17 E-2	0.17 E-2
	1.0	7 sig. figs.	0.21 E-7	0.12 E-7
$(1+25x^2)^{-1}$	0.1	2 sig. figs.	0.55 E-3	0.94 E-3
	0.1	7 sig. figs.	0.11 E-3	0.62 E-3
	1.0	2–7 figs.	0.59 E-1	0.55 E-1

But for low accuracy the data errors exceed the truncation errors in both methods, so both methods give about the same result.

5. Conclusions

It is found that spline interpolation is not as good as often supposed, when compared to Lagrangian methods. On the other hand, it has been claimed by some that splines are better because they are smoother, yet it seems to this author that if the underlying function is smooth, then the more accurate approximation will also be smoother.

References

[1] G.E. Forsythe, M.A. Malcolm and Č.B. Moler, *Computer Methods for Mathematical Computations* (Prentice-Hall, Englewood Cliffs, NJ, 1977).
[2] *I.M.S.L. Reference Manual* (I.M.S.L., Houston, Texas, 1980) Ch.1.
[3] A.J. MacLeod, A comparison of algorithms for polynomial interpolation, *J. Comput. Appl. Math.* **8** (1982) 275.
[4] L.F. Shampine and R.C. Allen, *Numerical Computing: An Introduction* (Saunders, Philadelphia, PA, 1973).