The Numerical Stability of Barycentric Lagrange Interpolation

Abhigyan Roy
Hrishav Das
Agni Ravi Deepa
Reva Dhillon

30th September 2024

AS5580 Assignment 2 Presentation (Group 3)

Contents:

- Why the need for alternate forms?
- The solution?
- Objective
- Notions of Stability
- Error Analysis
- 1st form
- 2nd form
- Numerical experiments
- Conclusion

Why the need for alternate forms?

The main reasons:

- **To reduce time complexity**, as the original formulation using the lagrange basis polynomials needs the computation of products of differences. This becomes an issue in bigger datasets as the complexity is $\mathbf{O}(n^2)$

Why the need for alternate forms?

The main reasons:

- **To reduce time complexity**, as the original formulation using the lagrange basis polynomials needs the computation of products of differences. This becomes an issue in bigger datasets as the complexity is $\mathbf{O(}n^2\mathbf{)}$
- Adding a new data point is costly as well since the lagrange polynomials need to be computed again.

Why the need for alternate forms?

The main reasons:

- **To reduce time complexity**, as the original formulation using the lagrange basis polynomials needs the computation of products of differences. This becomes an issue in bigger datasets as the complexity is $\mathbf{O}(n^2)$
- Adding a new data point is costly as well since the lagrange polynomials need to be computed again.
- Lagrange interpolation is numerically unstable especially if the node points are not chosen apporpriately.

The solution?

- To come up with alternate formulations of the lagrange interpolation methods that reduce the time complexity and are numerically stable.

The solution?

- To come up with alternate formulations of the lagrange interpolation methods that reduce the time complexity and are numerically stable.
- Two modified forms have been formulated, both being the **Barycentric forms** of Lagrange interpolation.

Objective

To show the numerical stability of both forms, more specifically we can show that,

Objective

To show the numerical stability of both forms, more specifically we can show that,

- The first form is unconditionally stable.

Objective

To show the numerical stability of both forms, more specifically we can show that,

- The first form is unconditionally stable.
- The second form is conditionally stable.

Objective

To show the numerical stability of both forms, more specifically we can show that,

- The first form is unconditionally stable.
- The second form is conditionally stable.

The error analysis has been done for both these forms and compared with the Newton divided difference form for two different functions.

Error Analysis and Stability

Let's define some preliminaries

- Condition Number
- Notions of Stability
- Lebesgue Constant
- Floating Point Arithmetic Model

Condition Number

The condition number of p_n at x with respect to f is defined for $p_n(x) \neq 0$:

$$\text{cond}(x, n, f) = \lim_{\Delta f \to 0} \sup \frac{|p_f(x) - p_{f+\Delta f}(x)|}{|\epsilon p_f(x)|}, \text{ where } |\Delta f| \leq \epsilon |f|. \tag{1}$$

In the notation, cond(x, n, f), the term 'n' indicates the dependence on the points x_j.

Condition Number

The condition number of p_n at x with respect to f is defined for $p_n(x) \neq 0$:

$$\text{cond}(x, n, f) = \lim_{\Delta f \to 0} \sup \frac{|p_f(x) - p_{f+\Delta f}(x)|}{|\epsilon p_f(x)|}, \text{ where } |\Delta f| \leq \epsilon |f|. \tag{1}$$

In the notation, cond(x, n, f), the term 'n' indicates the dependence on the points x_j.

$$\text{cond}(x, n, f) = \sum_{j=0}^{n} \left| \frac{\ell_j(x) f_j}{p_n(x)} \right| \geq 1, \tag{2}$$

and for any Δf with $|\Delta f| \leq \epsilon |f|$ we have

$$\frac{|p_f(x) - p_{f+\Delta f}(x)|}{|p_f(x)|} \leq \text{cond}(x, n, f) \cdot \epsilon \tag{3}$$

Notions of Stability: Forward Stable

A numerical method is **forward stable** if the error between **computed solution** and actual solution is small:

$$\text{Forward Error} = \frac{|\hat{f}(x) - f(x)|}{|f(x)|}$$

Notions of Stability: Forward Stable

A numerical method is **forward stable** if the error between **computed solution** and actual solution is small:

$$\text{Forward Error} = \frac{|\hat{f}(x) - f(x)|}{|f(x)|}$$

Forward Error Bound:

The forward error can be bounded as:

$$\frac{|\hat{f}(x) - f(x)|}{|f(x)|} \leq \text{cond}(x, n, f) \cdot u$$

where u is the **unit roundoff** (machine precision) and $\text{cond}(x, n, f)$ is the **condition number**.

Notions of Stability: Backward Stable

A numerical method is **backward stable** if the **computed solution** $\hat{f}(x)$ is the exact solution to a **perturbed problem** $f(x + \delta x)$, where δx is small.

Notions of Stability: Backward Stable

A numerical method is **backward stable** if the **computed solution** $\hat{f}(x)$ is the exact solution to a **perturbed problem** $f(x + \delta x)$, where δx is small.

Backward Error:

The **backward error** is defined as:

$$\text{Backward Error} = \min\{|\delta x| : f(x + \delta x) = \hat{f}(x)\}$$

A numerical method is **backward stable** if the **computed solution** $\hat{f}(x)$ is the exact solution to a **perturbed problem** $f(x + \delta x)$, where δx is small.

Backward Error:

The **backward error** is defined as:

$$\text{Backward Error} = \min\{|\delta x| : f(x + \delta x) = \hat{f}(x)\}$$

Backward Stability Condition:

A method is backward stable if:

$$|\delta x| \leq u \cdot |x|$$

meaning the computed result corresponds to a nearby problem.

Lebesgue Constant

We assume that the nodes, x_j lie in $[-1, 1]$ and express the bound in terms of Λ_n, the Lebesgue constant associated with the points x_j:

$$\Lambda_n = \sup_{f \in C([-1,1])} \frac{||P_n f||_\infty}{||f||_\infty} \tag{4}$$

where P_n is the operator mapping f to its interpolating polynomial at the points x_j,

$$||f|| = \max_{x \in [-1,1]} |f(x)|$$

and C is the space of all continuous functions from [-1,1]

Lebesgue Constant

We assume that the nodes, x_j lie in $[-1, 1]$ and express the bound in terms of Λ_n, the Lebesgue constant associated with the points x_j:

$$\Lambda_n = \sup_{f \in C([-1,1])} \frac{||P_n f||_\infty}{||f||_\infty} \tag{4}$$

where P_n is the operator mapping f to its interpolating polynomial at the points x_j,

$$||f|| = \max_{x \in [-1,1]} |f(x)|$$

and C is the space of all continuous functions from [-1,1]
It can be shown that:

$$\Lambda_n = \sup_{x \in [-1,1]} \sum_{j=0}^{n} |\ell_j(x)| \tag{5}$$

Floating-Point Arithmetic Model

Standard Model of Floating-Point Arithmetic (Higham, 2002):

$$\mathrm{fl}(x \,\mathrm{op}\, y) = (x \,\mathrm{op}\, y)(1+\delta)^{\pm 1}, \quad |\delta| \le u$$

where:

- $\mathrm{fl}(x \,\mathrm{op}\, y)$: result in floating-point arithmetic
- u: unit roundoff
- δ: rounding error, $|\delta| \le u$

This formula models how floating-point arithmetic introduces errors in operations like $+, -, *, /$.

Relative Error Accumulation

Relative Error Counter:

$$\langle k \rangle = \prod_{i=1}^{k} (1+\delta_i)^{\rho_i}, \quad \rho_i = \pm 1, \quad |\delta_i| \le u$$

This captures how errors accumulate over multiple operations.

Relative Error Accumulation

Relative Error Counter:

$$\langle k \rangle = \prod_{i=1}^{k} (1+\delta_i)^{\rho_i}, \quad \rho_i = \pm 1, \quad |\delta_i| \le u$$

This captures how errors accumulate over multiple operations.

Bound on Relative Error:

$$|\langle k \rangle - 1| \le \gamma_k = \frac{ku}{1-ku}$$

where k is the number of operations and u is the unit roundoff. The error grows as a function of ku, controlling the relative error in the worst case.

1st Form: Definition

Recall traditional Lagrange representation of an interpolating polynomial is written as:

$$p_n(x) = \sum_{j=0}^{n} \mathcal{L}_j(x) f_j \tag{6}$$

where $\mathcal{L}_j(x) = \prod_{k=0, k\neq j}^{n} \frac{x - x_k}{x_j - x_k}$

We can rewrite this expression as follows:

$$p_n(x) = \underbrace{\left(\prod_{k=0}^{n}(x - x_k)\right)}_{\ell(x)} \sum_{j=0}^{n} \left(\frac{1}{x - x_j}\right) \underbrace{\left(\prod_{k=0,k\neq j}^{n} \frac{1}{x_j - x_k}\right)}_{w_j} f_j \quad (7)$$

$$\boxed{p_n(x) = \ell(x) \sum_{j=0}^{n} \left(\frac{w_j}{x - x_j}\right) f_j} \quad (8)$$

The form presented in Eq.8 is called the **First Form of the Barycentric Interpolation Formula**.

Error Analysis on First Form:

- The computed weights w_j satisfy:

$$\hat{w}_j = w_j \langle 2n \rangle_j, \quad j = 0 : n \tag{9}$$

- The computed $\hat{l}(x)$ satisfies:

$$\hat{l}(x) = l(x) \langle 2n + 1 \rangle_j \tag{10}$$

- The computed interpolating polynomial $\hat{p}_n(x)$ satisfies:

$$\boxed{\hat{p}_n(x) = l(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j} f_j \langle 5n + 5 \rangle_j} \tag{11}$$

Error Analysis on the First Form:

- Eq.11 can be interpreted as the value of $\hat{p}_n(x)$ computed in presence of perturbations in f_i. Therefore, it is backward stable.
- We can also bound a forward error as follows:

$$\frac{\|p_n(x) - \hat{p}_n(x)\|}{\|p_n(x)\|} \leq (\gamma_{5N+5})\text{cond}(x, n, f) \tag{12}$$

- The First Form representation is shown to be **Backward Stable and Forward Stable**.

2nd Form: Definition

If the function values f_j are 1, they will be interpolated by $p_n(x) = 1$. From eq. 8, this gives,

$$\ell(x) = \frac{1}{\sum_{j=0}^{n} \left(\frac{w_j}{x - x_j} \right)} \tag{13}$$

Substituting the expression for $\ell(x)$ in eq. 8, we obtain:

$$\boxed{p_n(x) = \frac{\sum_{j=0}^{n} \left(\frac{w_j}{x - x_j} \right) f_j}{\sum_{j=0}^{n} \left(\frac{w_j}{x - x_j} \right)}} \tag{14}$$

The form presented in Eq.14 is called the **Second (proper) form of the barycentric formula**.

Error and Stability Analysis of the Second Form

- **Different stability properties expected**: This formula is obtained by using a mathematical identity that does not necessarily hold in floating point arithmetic.
- The computed interpolating polynomial is:

$$\hat{p}_n(x) = \frac{\sum_{j=0}^{n} \left(\frac{w_j}{x-x_j}\right) f_j \langle 3n+4 \rangle_j}{\sum_{j=0}^{n} \left(\frac{w_j}{x-x_j}\right) \langle 3n+2 \rangle_j} \tag{15}$$

- Now, from eq. 2,

$$\mathrm{cond}(x, n, f) = \frac{\sum_{j=0}^{n} \left|\frac{w_j f_j}{x-x_j}\right|}{\left|\sum_{j=0}^{n} \frac{w_j f_j}{x-x_j}\right|} \tag{16}$$

Error and Stability Analysis of the Second Form

- Similarly,

$$\text{cond}(x, n, 1) = \frac{\sum_{j=0}^{n} \left| \frac{w_j}{x - x_j} \right|}{\left| \sum_{j=0}^{n} \frac{w_j}{x - x_j} \right|} \tag{17}$$

- Theorem: The computed $\hat{p}_n(x)$ satisfies,

$$\frac{|p_n(x) - \hat{p}_n(x)|}{|p_n(x)|} \leq (3n+4)u \frac{\sum_{j=0}^{n} \left| \frac{w_j f_j}{x - x_j} \right|}{\left| \sum_{j=0}^{n} \frac{w_j f_j}{x - x_j} \right|} + (3n+2)u \frac{\sum_{j=0}^{n} \left| \frac{w_j}{x - x_j} \right|}{\left| \sum_{j=0}^{n} \frac{w_j}{x - x_j} \right|} + O(u^2)$$

$$= (3n+4)u\,\text{cond}(x, n, f) + (3n+2)u\,\text{cond}(x, n, 1) + O(u^2) \tag{18}$$

Error and Stability Analysis of the Second Form

- It can be observed that for suitable choice of the x_j the second term in eq. 18 can be made arbitrarily large.
- For further analysis, we see that the Lebesgue constant may be written as:

$$\Lambda_n = \sup_{x \in [-1,1]} \sum_{j=0}^{n} |\ell_j(x)| \tag{19}$$

- Therefore, the forward error bound is given by:

$$\frac{|p_n(x) - \hat{p}_n(x)|}{|p_n(x)|} \leq -(3n+4)u\mathrm{cond}(x, n, f) + (3n+2)u\Lambda_n + O(u^2) \tag{20}$$

Chebyshev Points

For the Chebyshev points of the first kind (the zeros of the degree $(n+1)$ Chebyshev polynomial) and the Chebyshev points of the second kind (the extreme points of the degree n Chebyshev polynomial):

$$\Lambda_n \leq \frac{2}{\pi} \log(n+1) + 1.$$

For other 'good' sets of points, Λ_n is also slowly growing.

Comparison with Equally Spaced Points

For equally spaced points, Λ_n grows exponentially at a rate proportional to:

$$\frac{2^n}{n \log n}.$$

Stability Variation with Node Distributions

- We conclude that while the barycentric formula is not forward stable in general, it can be significantly less accurate than the modified Lagrange formula only for a poor choice of interpolating points and special functions f.
- More specifically, for both sets of Chebyshev points, the barycentric formula is guaranteed to be forward stable - that is, it produces relative errors bounded by:

$$g(n)u \cdot \mathrm{cond}(x, n, f), \quad \text{with } g \text{ a slowly growing function of } n.$$

- The computational advantage of the barycentric formula over the modified Lagrange formula: Since the w_j appear linearly in both the numerator and the denominator, they can be rescaled ($w_j \leftarrow \alpha w_j$) to avoid overflow and underflow.

Numerical Experiments

- Computations performed in MATLAB with $u \approx 10^{-16}$.
- 30 equally spaced points x_j on $[-1, 1]$ (thus $n = 29$).
- Evaluated interpolant at 100 points on $[-1 + 10^3\epsilon, 1 - 10^3\epsilon]$.
- 'Exact' values obtained using 50-digit arithmetic with MATLAB's Symbolic Math Toolbox.

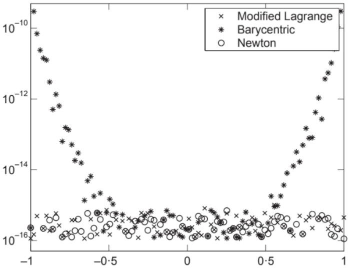
Function values for first case: $f_j = 0$ for $j = 0 : n-1$ and $f_n = 1$, $\mathrm{cond}(x, n, f) = 1$ and that $\Lambda_n = 3 \times 10^6$

Runge function for second case and $\mathrm{cond}(x, n, f) = 7.5$

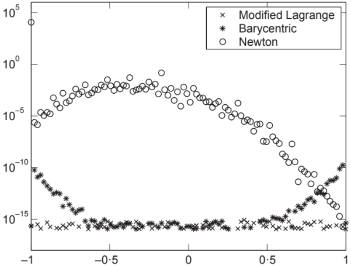
Observations from Numerical Experiments

- Modified Lagrange formula performs stably; barycentric formula performs unstably.
- Newton divided difference formula performs stably in increasing order but unstably in decreasing order.

N. J. HIGHAM



(a) x_i in increasing order.

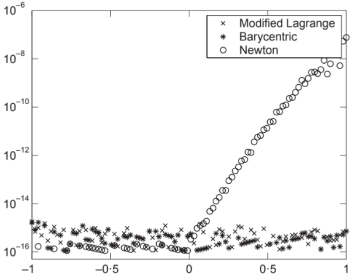


(b) x_i in decreasing order.

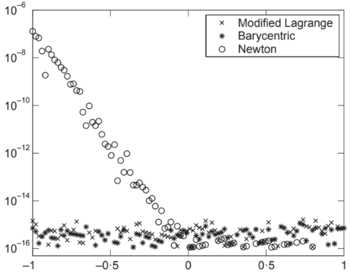
Figure 1: Comparison of Relative errors in computed $p_n(x)$ for 30 equally spaced points x_i

Further Experiments and Conclusions

- A second typical example with the Runge function $f(x) = \frac{1}{1+25x^2}$ and Chebyshev points.
- Modified Lagrange and barycentric formulas behave stably; Newton divided difference becomes unstable.



(a) Increasing order of Chebyshev points.



(b) Decreasing order of Chebyshev points.

Figure 2: Relative errors in computed $p_n(x)$ for 30 Chebyshev points of the first kind.

Conclusions

- It is evident that in all cases the the modified lagrange formula (first form of barycentric lagrange) is stable which implies unconditional stability.

[1] Higham, Nicholas J. / The numerical stability of barycentric Lagrange interpolation. In: IMA Journal of Numerical Analysis. 2004 ; Vol. 24, No. 4. pp. 547-556.

Conclusions

- It is evident that in all cases the the modified lagrange formula (first form of barycentric lagrange) is stable which implies unconditional stability.
- For the second form we can see see unstability for the dataset having a high Lebesgue constant. The requirement of a low Lebesgue constant for stability implies conditional stability.

[1] Higham, Nicholas J. / The numerical stability of barycentric Lagrange interpolation. In: IMA Journal of Numerical Analysis. 2004 ; Vol. 24, No. 4. pp. 547-556.

Conclusions

- It is evident that in all cases the the modified lagrange formula (first form of barycentric lagrange) is stable which implies unconditional stability.
- For the second form we can see see unstability for the dataset having a high Lebesgue constant. The requirement of a low Lebesgue constant for stability implies conditional stability.
- Both barycentric forms perfom significantly better than the Newton divided difference form.

Reference [1]

[1] Higham, Nicholas J. / The numerical stability of barycentric Lagrange interpolation. In: IMA Journal of Numerical Analysis. 2004 ; Vol. 24, No. 4. pp. 547-556.

Contributions

All team members contributed equally to the completion of this assignment and making of presentation, with active participation in research, analysis, and execution of tasks.