On a two-phase iterative-sandwich algorithm for improved polynomial approximation by a Modified Lupas Operator using statistical perspectives.

Robin Antoine, Letetia Addison & *Ashok Sahai

Department of Mathematics & Computer Science,

The University of West Indies; St. Augustine Campus @ TRINIDAD.

Abstract

This paper aims at constructing a two-phase iterative computerizable numerical algorithm for an improved approximation by '**Modified Lupas**' operator. The algorithm uses the 'statistical perspectives' for exploiting the information about the unknown function 'f' available in terms of its known values at the 'equidistant-knots' in C[0,1] more fully. The improvement, achieved by an aposteriori use of this information, happens iteratively. Any typical iteration uses the concepts of 'Mean Square Error (MSE)' and 'Bias' ; the application of the former being preceded by that of the latter in the algorithm. At any iteration, the statistical concept of 'MSE' is used in "Phase II", after that of the 'Bias' in "Phase I". Like a 'Sandwich', the top and bottom-breads are the operations of 'Bias-Reduction' per the "Phase I" of our algorithm, and the operation of 'MSE-Reduction' per the "Phase II" is the stuffing in the sandwich. The algorithm is an iterative one amounting to a desired-height 'Docked-Pile' of sandwiches with the bottom–bread of the first iteration serving as the top-bread for the second-iteration sandwich, and so-on-and-so forth. The potential of the achievable improvements through the proposed 'computerizable numerical iterative algorithm' is illustrated per an 'empirical study' for which the function 'f' is assumed to be known in the sense of simulation. The illustration has been confined to "Three Iterations" only, for the sake of simplicity of the illustration.

Keywords: Approximation; Modified Bernstein operator; Simulated empirical study.

AMS Classifications: **41A10, 41A36, 62F10.**

1. Introduction.

Szasz (1950) proposed the following generalization of the well-known Bernstein's polynomial approximation operator extending it to infinite interval.

$$S_n(f; x) = [e^{-nx} \sum_{k=0}^{\infty} \left\{\frac{(nx)^k}{k!}\right\} f(\tfrac{k}{n})], \text{for all } f \in C\,[0,\infty). \quad \textbf{(1.1)}$$

Heinz- Gerd Lehnoff (1984) proposed "Modified Szasz- Mirakjan Operator" as follows:

$$S_n(f; x) = \sum_{k=0}^{k=n} f\left(\frac{k}{n}\right) . T_k(x) / \sum_{k=0}^{k=\infty} T_k(x) . \forall\, x \in C[0,1]\ \&\ f \in C[0,1].$$

Wherein; $\mathbf{T_k(x) = (n.x)^k / k!}$ **(1.2)**

Motivated by that modification by **Heinz-Gerd Lehnhoff (1981),** but slightly differently, we propose a more appropriate modification of the **Lupas (1995) polynomial approximation operator**:

$$\mathbf{ML_n}\,(f; x) = \sum_{k=0}^{k=n} \binom{n+k-1}{k} * (2*x)^k \,.\mathbf{f}\left(\frac{2*k}{k+n}\right) / \left[\sum_{k=0}^{k=n} \binom{n+k-1}{k} * (2*x)^k\right];$$

For all f ε C [0, 1] & x ε C [0, 1] **(1.3)**

This modification is apparently a more-appropriate one, inasmuch as **"ML[n]"** could well be interpreted as **"Weighted- Average"** of the used **(n +1) known values of the unknown function "f(x)" , namely 'f(k/n)' ; k = 0 (1) n ; "Weights" being "$\mathbf{T_k(x)}$'s"**

;Wherein T_k (x) = $\binom{n+k-1}{k} * (2*x)^k$ **; x ε C[0,1]. In fact $T_k(x)$'s could be interpreted as proportional to "probabilities" ["$T_k(x) \geq 0$"].** As such, therefore,

$$ML[n] = E(f(x)) \qquad (1.4)$$

Incidentally, as we could use a suitable transformation (translation-n-change-of-scale) of the variable 'x', **we could assume, without loss of generality, that we are interested in the approximation of a bounded function f $\in C[0,1]$, even if the impugned function could, originally, be rather a bounded one f$\in C[a,b]$.**

2. Two-Phase Iterative-Sandwich Improvement Algorithm for Modified Lupas Operators ML[n].

In this section we propose the **"Two-Phase Iterative-Sandwich Improvement Algorithm for Modified Lupas Operators ML[n]",** using **'TWIN'** statistical perspectives of **'Bias' & 'MSE'**. In the statistical sense **'ML[n]'** is an estimate of the unknown function **'f(x)**$\sim\theta$'. Now, we use our estimator/modified Lupas Operators **'ML[n]' to estimate the values of the unknown function 'f(x)' at the knots '(k/n)', say Et f(k/n), k = 0 (1) n, Vis-$\grave{a}$-Vis known values of the unknown function "f(x)", namely 'f(k/n)' ; k=0(1)n.**

Hence the **"Knot-Wise Error", say Er f (k/n)** $\equiv Etf\left(\frac{k}{n}\right) - f\left(\frac{k}{n}\right), k = 0\ (1)n$ **could be generated to lead to the calibration of the "Error/Bias Polynomial Function",**

Say Er L_n (f; x) =

$$\sum_{k=0}^{k=n} T_k(x).Erf\left(\frac{k}{n}\right) / \sum_{k=0}^{k=n} T_k(x).$$

(2.1)

On the other hand, the **"Modified Lupas Polynomial" approximation/estimator of the unknown function 'f(x)' is "ML[n]", as per the equation (1.3) in the preceding section. This enables us to achieve per our "Phase I" of the iterative algorithm, the "Reduced-Bias Polynomial" approximation/estimator of the unknown function "f(x)" just by subtracting the "Estimated Error/Bias Polynomial" per (2.1) above, to get:**

Say O_n (f; x) = ML[n] – Er L_n (f; x) (2.2)

Now, we embark upon **the "Phase II"** of our proposed **'Iterative Algorithm'**. The concept "**Minimum Mean Square Error Estimator (MMSEE)"** of **Searles (1964)** is seminal to this phase of our algorithm. As per **(1.4),** our **"Modified Lupas Polynomial" estimator** is rather analogous to **the sample-mean '$\bar{x}$'. Searles (1964)** considered **the class of estimators 'k. $\bar{x}$ ' ,** and chose the "Optimal" value say **"k_0"** by minimizing the **MSE (k. $\bar{x}$) to lead to the MMSEE $\sim$ "k_0. $\bar{x}$" Analogously, we consider the perturbed 'Polynomial', say b. O_n (f; x), and hence determine the estimated** values of the unknown function 'f(x)' at the knots '(k/n)', **say Et f (k/n) , k = 0 (1) n, vis-a-vis known values of the unknown function "f(x)", namely 'f(k/n)' ; k = 0 (1) n.**

Hence the **"Knot-Wise Squared-Error", say E2rf (k/n) ≡ [b*Et f (k/n) – f (k/n)]^2, k = 0 (1) n could be generated to lead to the calibration of the "Squared-Error Polynomial Function",**

Say E2r = $\sum_{k=0}^{k=n} E2rf\left(\frac{k}{n}\right)$

(2.3)

This will be a "Quadratic Polynomial in b", say Q (b) ≡ A.b^2 + B. b + C. To avoid any complex solution to Q (b) = 0, we chose b_0 = - (B/2.A) to minimize the value of the MSE, leading to a 'Reduced-MSE Polynomial' estimator "b_0. O_n (f; x)".

To complete the **"First Iteration"** we again apply the details of the **'Phase I'** to treat our **"Reduced-MSE Polynomial" estimator "b_0. O_n (f; x)", to achieve the improved [at Iteration #1] 'Modified Lupas Polynomial' Operator/Estimator; Say:**

I [#1] ML[n] ≡ [Reduced- Bias Version Using 'Phase I' (Iteration #1) on "b_0. O_n (f; x)"] (2.4)

Thus operations defining **"FIRST Iteration"** could well be characterized as a **"Sandwich"!** The top and bottom-breads are **the operations of 'Bias-Reduction' per the "Phase I" of our algorithm, and the operation of the 'MSE-Reduction' per the "Phase II" is the stuffing in the sandwich.** The algorithm is iterative one, amounting **to the desired-height 'Docked-Pile' of sandwiches** with the **bottom-bread** of the **first iteration** serving as **the top-bread for the second-iteration sandwich**, and so-on-and-so forth.

At any iteration, the improvements will begin-and-end with the "Phase I" operation, sandwiching its "Phase II" operation of the 'MSE-

Reduction'. As such, at any iteration, we will have two improvement-operations only, namely that of 'Phase II' followed by that of 'Phase I' borrowing the last operation of the 'Bias-Reduction' per the 'Phase I' in the preceding iteration. Only the "First Iteration" will be an exception using three improvement-operations ~ Phase I – Phase II - Phase I.

3. The empirical simulation study.

To illustrate the gain in the efficiency of the "Modified Lupas Operators" by using our proposed "Sandwich- Iterative Algorithm of Improvement of Polynomial Approximation", we have carried an empirical study. We have taken **the example-cases of n = 3, 4, and 5 (i.e. n + 1 = 4, 5, and 6, knots)** in the empirical study to numerically illustrate the relative gain in efficiency in using the Algorithm Vis-à-Vis the Original Modified Lupas Polynomial Operator in each example-case of the n-values. Essentially, the empirical study is a simulation one wherein we would have to assume that the function, being tried to be approximated, namely f(x) is known to us. Once again, we have confined to the illustrations of the relative gain in efficiency by the Iterative Improvement for the following four illustrative functions: **f (x) = exp(x), ln (2 + x), sin (2+x*π/2), & 2^x.**

To illustrate the **POTENTIAL** of the improvement with our proposed Sandwich-Iterative Algorithm, we have considered **THREE Iterations**, and the numerical values of **seven – quantities: including the three Percentage Relative Errors (PREs)**, corresponding to our **Improvement Iteration (# = 1, 2, or 3) (PRE_I (#) ML_n (f ;x)) [n]), following that for Original Modified Lupas Polynomial Operator (PRE_ML_n (f; x) [n]), and the three corresponding Percentage Relative Gains (PRGs)** in using our Iterative Algorithmic Modified Lupas Polynomial Operators in place of the Original Modified Lupas Polynomial Operators **ML_n [n], namely (PRG_I (#) ML_n [n]; # 1 (1) 3).**

These quantities are defined, as follows.

PRE using original Modified Lupas (Polynomial) using n intervals in [0, 1], i.e. [(k- 1)/n, k/n]; k = 1 (1) n:

$$\text{PRE\_ MLn [n]} = \frac{\left|\int_0^1 f(x)dx - \int_0^1 ML_n(n)dx\right|}{\int_0^1 f(x)dx} x\ 100.$$

The PRE using Improvement Iteration (I #1, or 2, or 3) on Lupas (Polynomial) Operator using n intervals in [0, 1], i.e. [(k-1)/n, k/n]; k = 1 (1) n:

$$\text{PRE\_I}(\#)ML_n[n] = \frac{\left|\int_0^1 f(x)dx - \int_0^1 I(\#)ML_n(n)dx\right|}{\int_0^1 f(x)dx} x\ 100; \# = 1; \text{or } 2; \text{or } 3.$$

The **PREs** respective to the **Original Modified Lupas Polynomial Operator, and respective to the First, Second and Third Sandwich-Algorithmic Improvement Iteration Polynomials, respectively, for each of the example # of approximation Knots/Intervals and the Percentage Relative Gains (PRGs), defined exactly analogously to PREs, by using the proposed Sandwich-Algorithmic Improvement Iteration: I# [e.g. 1 , or 2 , or 3] Polynomials with the n intervals in [0, 1] over using the Original Modified Lupas Polynomial Operator for the approximation of the Targeted function, 'f (x)' are tabulated in the following four tables:~ Tables 1 to Table 4.**

4. Conclusion.

The tabulated values of PRGs in the "APPENDIX" amply illustrate the 'Relative Gains by using the proposed "Two-phase iterative algorithm using the statistical perspectives for improved approximation by Modified Lupas Operator". Even for 6 knots (n = 5), the PRGs are above 90% for all example-functions, after only THREE iterations. For f(x) = sin (2+x*π/2) & ln (2+x) they are above 99%, after the three iterations!

References/Bibliography.

[1] N.L. Carothers, A Short Course on Approximation Theory, Bowling Green State University, Bowling Green, OH, 1998.

[2] N.L. Carothers, Real Analysis, Cambridge University Press, 2000.

[3] W. Cheney, D. Kincaid, Numerical Mathematics and Computing, Brooks/Cole Publishing Company,1994.

[4] P.J. Hartley, A. Wynn-Evans, A Structured Introduction to Numerical Mathematics, Stanley Thornes, 1979.

[5] E.R. Hedrick, The significance of Weirstrass theorem, The Amer. Math. Monthly 20 (1927) 211–213.

[6] **Heinz-Gerd Lehnhoff**, On a Modified Szasz-Mirakjan Operator. J. Approx. Theory. 42: 278-282 (**1984**).

[7] G.G. Lorentz, Approximation of Functions, Chelsea, 1986.

[8] A. Lupas, The approximation by means of some linear positive operators, in: Approximation Theory (Proceedings of the International Dortmund Meeting IDoMAT 95, held in Witten, Germany, March 13-17, 1995), M. W. M¨uller, M. Felten, and D. H. Mache, eds. (Mathematical Research, Vol. 86) Akademie Verlag, Berlin 1995, pp.201-229.

[9] B.F. Polybon Applied Numerical Analysis, PWS-KENT, 1992.

[10] A. Sahai, An iterative algorithm for improved approximation by Bernstein's operator using statistical

perspective, Applied Mathematics and Computation 149 (2004) 327-335.
[11] D.T. Searles, The utilization of known coefficient of variation in the estimation procedure, JASA. Vol. 59, 1964
[12] A. Sheilds, Polynomial Approximation, The Math. Intell. 9 (3) (1987)5–7.
[13] K. Weierstrass, Uber die analytische Darstellbarkeit sogenannter willkurlicher Functionen einer reellen Veranderlichen Sitzungsberichteder Koniglich Preussischen Akademie der Wissenshcaften zu Berlin, 1885, pp. 633–639, 789–805.

APPENDIX.

Table 1 :(Iterative) Algorithmic (In %) Relative Efficiency/ Gain for f (x) = exp(x).

Items ↓	n→ 3	4	5
PRE_ ML_n (f; x) [n]	24.99015819	28.78914401	31.2438483
PRE_I (1) ML_n (f; x) [n]	14.52845941	14.42598420	15.0358634
PRE_I (2) ML_n (f; x) [n]	10.62695956	9.94569919	10.3341992
PRE_I (3) ML_n (f; x) [n]	3.17654665	1.23138525	0.8511202
PRG_I (1) ML_n (f; x) [n]	41.86327554	49.89088875	51.87576340
PRG_I (2) ML_n (f; x) [n]	57.47542101	65.45330005	66.92405144
PRG_I (3) ML_n (f; x) [n]	87.28880932	95.72274448	97.27587888

Table 2: (Iterative) Algorithmic (In %) Relative Efficiency/Gain for f (x) = ln (2+x).

Items ↓	n→ 3	4	5
PRE_ ML_n (f; x) [n]	8.52209478	10.2707945	11.31455004
PRE_I (1) ML_n (f; x) [n]	5.30666820	5.3304237	5.52294491
PRE_I (2) ML_n (f; x) [n]	3.90199321	3.5226194	3.48653274
PRE_I (3) ML_n (f; x) [n]	1.44370539	0.4311038	0.00538581
PRG_I (1) ML_n (f; x) [n]	37.73047191	48.10115463	51.18723328
PRG_I (2) ML_n (f; x) [n]	54.21321500	65.70256111	69.18540523
PRG_I (3) ML_n (f; x) [n]	83.05926620	95.80262443	99.95239917

Table 3: (Iterative) Algorithmic (In %) Relative Efficiency/Gain for f (x) = sin (2+x*π/2).

Items ↓	n→ 3	4	5
PRE_ ML_n (f; x) [n]	101.62048355	117.93489106	128.21083454
PRE_I (1) ML_n (f; x) [n]	60.61951304	60.74273117	65.24809523
PRE_I (2) ML_n (f; x) [n]	42.50271503	38.73711552	33.08623891
PRE_I (3) ML_n (f; x) [n]	21.47465476	14.27798701	11.87840116
PRG_I (1) ML_n (f; x) [n]	40.34715153	48.49469003	49.10875084
PRG_I (2) ML_n (f; x) [n]	58.17505136	67.15381242	74.19388226
PRG_I (3) ML_n (f; x) [n]	78.86778923	87.89333091	90.73525943

Table 4: (Iterative) Algorithmic (In %) Relative Efficiency/Gain for $f(x) = 2^x$.

Items ↓	**n→ 3**	**4**	**5**
PRE_ ML_n (f; x) [n]	**16.50183845**	**19.15625639**	**20.84201688**
PRE_I (1) ML_n (f; x) [n]	**9.61646599**	**9.46970940**	**9.79775441**
PRE_I (2) ML_n (f; x) [n]	**6.89370311**	**6.22104973**	**6.29499328**
PRE_I (3) ML_n (f; x) [n]	**1.90171541**	**0.29965161**	**0.18722834**
PRG_I (1) ML_n (f; x) [n]	**41.72488101**	**50.56597065**	**52.99037293**
PRG_I (2) ML_n (f; x) [n]	**58.22463578**	**67.52471045**	**69.79662129**
PRG_I (3) ML_n (f; x) [n]	**88.47573605**	**98.43575063**	**99.10167840**