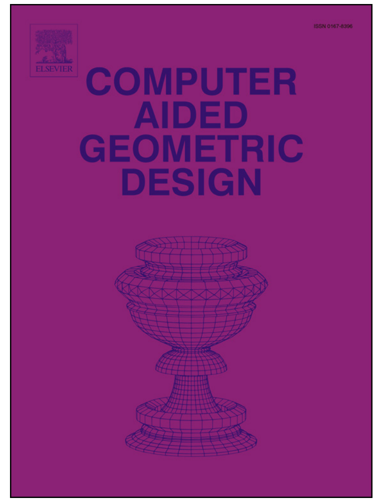
Accepted Manuscript



A note on zeros of univariate scalar Bernstein polynomials

Jinesh Machchhar, Gershon Elber

PII: S0167-8396(18)30105-5
DOI: https://doi.org/10.1016/j.cagd.2018.09.001
Reference: COMAID 1709

To appear in: *Computer Aided Geometric Design*

Received date: 12 April 2018
Revised date: 30 August 2018
Accepted date: 4 September 2018

Please cite this article in press as: Machchhar, J., Elber, G. A note on zeros of univariate scalar Bernstein polynomials. *Comput. Aided Geom. Des.* (2018), https://doi.org/10.1016/j.cagd.2018.09.001

A Note on Zeros of Univariate Scalar Bernstein polynomials

Jinesh Machchhar and Gershon Elber

September 7, 2018

Abstract

In [3], an algorithm is presented for computing all real roots of univariate scalar Bernstein polynomials by subdividing the polynomial at a known root and then factoring out the root from the polynomial, resulting in a reduction in problem complexity. This short report presents a speed-up over [3], by circumventing the need for subdividing the polynomial each time a root is discovered, an $O(n^2)$ process, where n is the order of the polynomial. The subdivision step is substituted for by a polynomial division. This alternative also has some drawbacks which are discussed as well.

Keywords: Polynomial roots; Bernstein polynomials; Zero-set; Bernstein basis; Polynomial division

1 Introduction and related work

Computing zero-sets of polynomials is an ubiquitous problem in numerous fields in the sciences, engineering, and design. Applications in geometric design are many and varied: implicit representation of manifolds, computing intersections of manifolds, sweeps and offsets, to name a few. This short note aims to further improve upon a recently proposed method [3] for computing the roots of univariate scalar Bernstein polynomials. Unlike traditional approaches [4, 5] which subdivide the polynomial at arbitrary locations, in [3], the polynomial is subdivided at a known root.

The algorithm proposed in this paper continues the line of thought in [3], while replacing the subdivision of the input polynomial at a known root t_0, with a polynomial division by $(t - t_0)$. Each time a root is discovered, it is factored out using a linear-time routine, which works by reversing the process of Bernstein polynomial multiplication, and is discussed in Section 2. The overall algorithm for computing real roots is given in Section 3. A comparison of running times with the previous state-of-the-art method in [3] over a large number of polynomials, is performed in Section 4, and shows a speed-up by a factor of about two, especially on polynomials of lower degrees. This report concludes in Section 5, with remarks on issues related to numerical stability and computational gain.

2 Factoring out roots

Let $c(t)$ be a univariate scalar Bernstein polynomial of degree $m+1$, expressed as,

$$c(t) = \sum_{i=0}^{m+1} c_i B_{i,m+1}(t), \tag{1}$$

where $B_{i,m+1}, 0 \leq i \leq m+1$, are the Bernstein basis functions and $c_i, 0 \leq i \leq m+1$, are the Bernstein coefficients. Suppose that $c(t)$ has a known root at $t_0 \in [0, 1]$, i.e., $c(t_0) = 0$. By the fundamental theorem of the algebra, $c(t)$ may be expressed as a product of two polynomials, viz., $r(t)(t - t_0)$, where $r(t)$ is of degree m. Given a polynomial $r(t) = \sum_{i=0}^{m} r_i B_{i,m}(t)$ of degree m with Bernstein coefficients $r_i, i = 0, \ldots, m$ and a polynomial $g(t) = \sum_{i=0}^{k} g_i B_{i,k}(t)$ of degree k with Bernstein coefficients $g_i, i = 0, \ldots, k$, their product, $c(t) = r(t)g(t)$ is expressed in Bernstein form as follows [2]:

$$c(t) = \sum_{i=0}^{m+k} \left\{ \sum_{j=\max(0,i-k)}^{\min(m,i)} \frac{\binom{m}{j}\binom{k}{i-j}}{\binom{m+k}{i}} r_j g_{i-j} \right\} B_{i,m+k}(t), \tag{2}$$