An Efficient Algorithm for Evaluating Polynomials in the Pólya Basis

J. Warren, Houston

Abstract. A new $O(n)$ algorithm is given for evaluating univariate polynomials of degree $n$ in the Pólya basis. Since the Lagrange, Bernstein, and monomial bases are all special instances of the Pólya basis, this technique leads to efficient evaluation algorithms for these special bases. For the monomial basis, this algorithm is shown to be equivalent to Horner’s rule.

Key words: Polynomial bases, evaluation.

1. Pólya Basis Functions

Let $n_k(t)$ be the restriction of the standard B-spline basis functions [3] of degree $n$ over the nondecreasing knot sequence $t_1, t_2, \ldots, t_{2n}$ to the interval $t_n < t < t_{n+1}$. The Pólya basis functions are the unique polynomial functions $d_k(t)$ that satisfy Marsden’s identity [5].

$$(x - t)^n = \sum_{k=0}^{n} d_k(t)n_k(x).$$

The Pólya basis functions of degree $n$ can be written explicitly as

$$d_k(t) = \prod_{i=1}^{n} (t_{k+i} - t). \quad (1)$$

(See Barry and Goldman ([2], pp. 28) for more details.)

Using the explicit definition of Eq. (1), the Pólya functions can be defined directly for arbitrary sequences $t_1, \ldots, t_{2n}$. Under this definition, the Pólya functions includes several important bases as special cases. These bases are:

- The Lagrange basis,
- The Bernstein basis,
- The monomial basis.

Given a set of $n + 1$ distinct values $u_0, \ldots, u_n$, The Lagrange basis functions $l_k(t)$ [1] of degree $n$ are the $n + 1$ polynomial basis functions that satisfy

$$l_k(u_i) = \delta_{ki}.$$  

The $k$th Lagrange basis function can be explicitly written as

$$l_k(t) = \prod_{i \neq k} \frac{1}{x_k} (u_i - t)$$
where \( z_k = \prod_{i \neq k} (u_i - u_k) \). The Lagrange basis may be viewed as an instance of the Pólya basis. Consider the Pólya basis with knot sequence

\[
\begin{align*}
t_1 &= u_1, t_2 = u_2, \ldots, t_n = u_n, \\
t_{n+1} &= u_0, t_{n+2} = u_1, \ldots, t_{2n} = u_{n-1}.
\end{align*}
\]

The Pólya basis function \( d_k(t) \) is exactly the Lagrange basis function \( l_k(t) \) multiplied by the constant \( z_k \).

The Bernstein basis functions of degree \( n \) are

\[
b_k(t) = \frac{n!}{k!(n-k)!} t^k (1-t)^{n-k}.
\]

These basis functions are the building blocks for one of the most popular curve representations in geometric design, Bézier curves [4]. Specializing the knot sequence to

\[
\begin{align*}
t_1 &= 0, t_2 = 0, \ldots, t_n = 0, \\
t_{n+1} &= 1, t_{n+2} = 1, \ldots, t_{2n} = 1
\end{align*}
\]

yields a Pólya basis of the form

\[
d_k(t) = (-t)^{n-k} (1-t)^k.
\]

Thus, the Pólya basis function \( d_k(t) \) is exactly the Bernstein basis function \( b_{n-k}(t) \) multiplied by the constant \( (-1)^{n-k} \frac{n!}{k!(n-k)!} \).

Finally, the monomial basis, \( \{1, t, t^2, \ldots, t^n\} \) is probably the most commonly used basis in mathematics. To express the monomial basis as a Pólya basis, one must first develop a notion of a knot at infinity. The affine knot \( t_i \) can be homogenized to yield the homogeneous representation \( (t_i, 1) \). The linear factors in the Pólya basis can be expressed in the form \( (t_i-1)*1 - 1*t) \). If \( t_i = \infty \), then the corresponding homogeneous knot is \( (1, 0) \). The related linear factor is \( (1 * 1 - 0 * t) \), the constant 1. (For information on this interpretation, see [2].) Thus, specializing the knot sequence to have values

\[
\begin{align*}
t_1 &= 0, t_2 = 0, \ldots, t_n = 0, \\
t_{n+1} &= \infty, t_{n+2} = \infty, \ldots, t_{2n} = \infty
\end{align*}
\]

yields Pólya basis functions of the form

\[
d_k(t) = t^{n-k}.
\]

These basis functions are exactly the monomial basis function after renumbering.

2. Evaluation in the Pólya Basis

Let \( d(t) \) be a polynomial of degree \( n \) in the Pólya basis.

\[
d(t) = \sum_{k=0}^{n} P_k d_k(t).
\]

(2)