A COMBINATORIAL CONSTRUCTION OF HIGH ORDER ALGORITHMS FOR FINDING POLYNOMIAL ROOTS OF KNOWN MULTIPLICITY

YI JIN AND BAHMAN KALANTARI

(Communicated by Peter A. Clarkson)

Abstract. We construct a family of high order iteration functions for finding polynomial roots of a known multiplicity $s$. This family is a generalization of a fundamental family of high order algorithms for simple roots that dates back to Schröder’s 1870 paper. It starts with the well known variant of Newton’s method $\hat{B}_2(x) = x - s \cdot p(x)/p'(x)$ and the multiple root counterpart of Halley’s method derived by Hansen and Patrick. Our approach demonstrates the relevance and power of algebraic combinatorial techniques in studying rational root-finding iteration functions.

1. Introduction

In [4] we introduced symmetric functions to the study of iterative root-finding algorithms, thus revealing the combinatorial nature of rational root-finding iteration functions. In particular, we gave a combinatorial interpretation of the Basic Family - a fundamental family of high order algorithms for finding simple roots of a polynomial. The Basic Family first appeared in Schröder’s 1870 paper [14] and was later rediscovered many times under different representations. In one representation, it is known as König’s family (see [1]). In [7], Kalantari et al. gave a purely algebraic derivation of the Basic Family and revealed many interesting minimality and uniqueness properties of this family. For some related works in this direction, see [8], [9], [10], [11] and [6].

In this article we demonstrate the power of our combinatorial approach by deriving an efficient algorithm to construct a generalized Basic Family for roots with a known multiplicity. Our exposition is organized as follows: In Section 2 we give a brief review of relevant results from [4]. In Section 3 we present a combinatorial construction of the generalized Basic Family and prove its convergence property. In Section 4 we show the connection between the Basic Family and its generalization.
2. Preliminaries

First we recall some notation and results from the theory of symmetric functions. For a definitive treatment on this subject, see MacDonald [13].

For each integer \( r \geq 1 \), the \( r \)-th elementary symmetric function \( e_r \) is the sum of all products of \( r \) distinct variables \( x_i \), the \( r \)-th complete symmetric function \( h_r \) is the sum of all monomials of total degree (the sum of exponents of all its variables) \( r \) in the variables \( x_1, x_2, \ldots \), and the \( r \)-th power sum is the sum of \( r \)-th power of \( x_1, x_2, \ldots \):

\[
e_r = \sum_{i_1 < i_2 < \cdots < i_r} x_{i_1} x_{i_2} \cdots x_{i_r},
\]

\[
h_r = \sum_{a_1 + a_2 + \cdots = r} x_1^{a_1} x_2^{a_2} \cdots ,
\]

\[
q_r = \sum_{i \geq 1} x_i^r.
\]

Additionally, define \( e_0 = h_0 = 1 \).

The generating functions for \( e_r, h_r \) and \( q_r \) are

\[
E(t) = \sum_{r \geq 0} e_r t^r = \prod_{i \geq 1} (1 + x_i t),
\]

\[
H(t) = \sum_{r \geq 0} h_r t^r = \prod_{i \geq 1} (1 - x_i t)^{-1},
\]

\[
Q(t) = \sum_{r \geq 1} q_r t^{r-1} = \sum_{i \geq 1} \frac{x_i}{t - x_i t}.
\]

Thus, \( H(t) = E(-t)^{-1} \) and

\[
Q(t) = \frac{d}{dt} \log H(t) = \frac{H'(t)}{H(t)} = \frac{E'(-t)}{E(-t)},
\]

where \( E'(-t) \) denotes the derivative of \( E \) evaluated at \(-t\).

The above relations among generating functions imply the following recursions:

\[
\sum_{r=0}^{n} (-1)^r e_r h_{n-r} = 0 \quad \text{for} \quad n = 1, 2, \ldots ,
\]

\[
\sum_{r=1}^{n} q_r h_{n-r} = n h_n \quad \text{for} \quad n = 1, 2, \ldots ,
\]

\[
\sum_{r=1}^{n} (-1)^{r-1} q_r e_{n-r} = n e_n \quad \text{for} \quad n = 1, 2, \ldots .
\]

We now make a connection between symmetric functions and root-finding algorithms.

Let \( p(x) \) be a polynomial of degree \( n \) with complex coefficients and let \( \theta_1, \ldots, \theta_n \) be its complex roots. Then,

\[
p(x) = c \prod_{i=1}^{n} (x - \theta_i),
\]

where \( c \) is the leading coefficient of \( p(x) \).

Define \( r_j = 1/(x - \theta_j), \ j = 1, \ldots, n \). Then we have
**Proposition 2.1** (Jin and Kalantari [4], Lemma 3.1).

\[ p^{(i)}(x) = i! p(x) e_i(r_1, \ldots, r_n), \]

where \( e_i \) is the \( i \)-th elementary symmetric function.

For each integer \( m \geq 2 \), define the \( m \)-th order Basic Family iteration function

\[ B_m(x) = x - \frac{h_{m-2}(r_1, \ldots, r_n)}{h_{m-1}(r_1, \ldots, r_n)}. \]

This is equivalent to the original definition of Basic Family given by Kalantari et al. [7].

**Proposition 2.2** (Jin and Kalantari [4], Corollary 3.4).

\( B_m(x) \) has an \( m \)-th order of convergence to a simple root and linear convergence to a multiple root.

Proposition 2.1 and the recurrence relation between \( h_r \) and \( e_r \) give rise to an efficient algorithm to compute \( B_m \).

**Algorithm 1**: Evaluation of the \( m \)-th order iteration function \( B_m \).

**input**: \( m, p(x), \) and \( x_0 \).

**output**: \( B_m(x_0) \).

\[ h[0] = 1; \]

for \( i \) from 1 to \( m-1 \) do

\[ e[i] = p^{(i)}(x_0) / (i! p(x_0)); \]

end

for \( i \) from 1 to \( m-1 \) do

\[ h[i] = \sum_{r=0}^{i-1} (-1)^{i-r-1} e[i-r] h[r]; \]

end

output \( B_m(x_0) = x_0 - h[m-2]/h[m-1]; \)

The first three members of the Basic Family are

\[ B_2(x) = x - \frac{p(x)}{p'(x)} \]

which is the well known Newton’s method,

\[ B_3(x) = x - \frac{2p(x)p'(x)}{2p'(x)^2 - p(x)p''(x)} \]

which is Halley’s method, and the 4th order method

\[ B_4(x) = x - \frac{6p(x)p'(x)^2 - 3p(x)^2p''(x)}{p(x)^2p''(x) + 6p'(x)^3 - 6p(x)p'(x)p''(x)} \]

For a root of multiplicity \( s \geq 2 \), there is a well known generalization of the 2nd order Newton’s method:

\[ \hat{B}_2(x) = x - s \frac{p(x)}{p'(x)}. \]

A generalization of the 3rd order Halley’s method was derived by Hansen and Patrick [3, Equation (8.2)].

In this article, we shall show that such generalization exists for all Basic Family iteration functions, and it can be computed efficiently due to its simple combinatorial characterization.
3. \( \{ \hat{B}_m \}_{m=2}^{\infty} - A FAMILY OF HIGH ORDER METHODS \)

FOR ROOTS OF MULTIPlicity \( s \)

The key ingredient in our combinatorial construction is a generalization of power sums and complete symmetric functions.

Define the generalized power sums

\[
\hat{q}_r = \frac{q_r}{s} = \frac{1}{s} \sum_{i \geq 1} x_i^r \quad \text{for} \quad r = 1, 2, \ldots
\]

The generalized complete symmetric functions \( \hat{h}_r \) are defined through the following recursion:

\[
\hat{h}_0 = 1, \quad \hat{h}_r = \frac{1}{r} \sum_{i=0}^{r-1} \hat{q}_{r-i} \hat{h}_i \quad \text{for} \quad r = 1, 2, \ldots
\]

Define the \( m \)-th order generalized Basic Family iteration functions

\[
\hat{B}_m(x) = x - \frac{\hat{h}_{m-2}(r_1, \ldots, r_n)}{\hat{h}_{m-1}(r_1, \ldots, r_n)} \quad \text{for} \quad m = 2, 3, \ldots
\]

Note that when \( s = 1 \), \( \hat{B}_m \) reduces to \( B_m \), an \( m \)-th order method for simple roots.

In the following subsection, we shall show that \( \hat{B}_m \) is an \( m \)-th order method for roots of multiplicity \( s \).

3.1. **Proof of convergence property of \( \hat{B}_m \).** Recall that the falling factorial is defined as

\[
(x)_0 = 1 \quad \text{and} \quad (x)_n = x(x-1) \ldots (x-n+1) \quad \text{for} \quad n = 1, 2, \ldots
\]

We first express \( \hat{h}_k(r_1, \ldots, r_n) \) as a polynomial in \( r_1 \).

**Lemma 3.1.** Assume that \( \theta_1 \) is a root of multiplicity \( s' \), i.e., \( \theta_1 = \theta_2 = \cdots = \theta_{s'} \), and \( \theta_i \neq \theta_1 \) for \( i = s' + 1, \ldots, n \). Define \( r_j = 1/(x - \theta_j) \), \( j = 1, \ldots, n \). Then for every integer \( k \geq 0 \),

\[
\hat{h}_k(r_1, \ldots, r_n) = \frac{k}{(k-1)!} \sum_{i=0}^{k} \frac{(k-i+s'/s-1)_{k-i}}{(k-i)!} \hat{h}_{i}(r_{s'+1}, \ldots, r_n) r_1^{k-i}.
\]

**Proof.** By induction on \( k \).

For \( k = 0 \), (3.4) holds trivially. Now assume that (3.4) holds for \( k < m \), \( m \in \mathbb{Z}^+ \).

Then for \( k = m \), by (3.2) we have the following recursion:

\[
\hat{h}_m(r_1, \ldots, r_n) = \frac{1}{m} \sum_{j=0}^{m-1} \hat{q}_{m-j}(r_1, \ldots, r_n) \hat{h}_j(r_1, \ldots, r_n).
\]

By (3.1) and induction assumption, we have

\[
\hat{q}_{m-j}(r_1, \ldots, r_n) = \frac{s'}{s} r_1^{m-j} + \hat{q}_{m-j}(r_{s'+1}, \ldots, r_n) \quad \text{for} \quad m - j > 0,
\]

\[
\hat{h}_j(r_1, \ldots, r_n) = \frac{1}{(j-1)!} \sum_{i=0}^{j} \frac{(j-i+s'/s-1)_{j-i}}{(j-i)!} \hat{h}_{i}(r_{s'+1}, \ldots, r_n) r_1^{j-i} \quad \text{for} \quad j < m.
\]
Plugging the above identities into (3.5), multiplying out, and collecting like terms of powers of \( r_1 \), we get

\[
\hat{h}_m(r_1, \ldots, r_n) = \frac{1}{m} \sum_{i=0}^{m} c_i r_1^{m-i},
\]

where

\[
c_i = \frac{s'}{s} \hat{h}_i(r_{s'+1}, \ldots, r_n) \sum_{j=i}^{m-1} \frac{(j-i+s'/s-1)_{j-i}}{(j-i)!} + \frac{(m-i+s'/s-1)_{m-i}}{(m-i)!} \sum_{j=1}^{i} \hat{h}_{i-j}(r_{s'+1}, \ldots, r_n) \hat{q}_j(r_{s'+1}, \ldots, r_n) = \frac{(m-i+s'/s-1)_{m-i}}{(m-i)!} \hat{h}_i(r_{s'+1}, \ldots, r_n)
\]

\[
= m \frac{(m-i+s'/s-1)_{m-i}}{(m-i)!} \hat{h}_i(r_{s'+1}, \ldots, r_n).
\]

Thus,

\[
\hat{h}_m(r_1, \ldots, r_n) = \sum_{i=0}^{m} \frac{(m-i+s'/s-1)_{m-i}}{(m-i)!} \hat{h}_i(r_{s'+1}, \ldots, r_n) r_1^{m-i}.
\]

So (3.4) holds for \( k = m \). Also, by induction, it holds for all \( k \geq 0 \). \( \square \)

Using the above structural lemma about \( \hat{h}_k \), it is easy to derive the convergence property of \( \hat{B}_m \).

**Theorem 3.2.** Let \( s' \) be the multiplicity of root \( \theta_1 \), i.e., \( \theta_1 = \theta_2 = \cdots = \theta_{s'} \), and \( \theta_i \neq \theta_1 \) for \( i = s'+1, \ldots, n \). Define \( r_j = 1/(x - \theta_j) \), \( j = 1, \ldots, n \).

When \( s' = s \), \( \hat{B}_m(x) \) has an order of convergence \( m \) for \( \theta_1 \), and

\[
\lim_{x \to \theta_1} \frac{\hat{B}_m(x) - \theta_1}{(x - \theta_1)^m} = \hat{h}_{m-1}((\theta_1 - \theta_{s'+1})^{-1}, \ldots, (\theta_1 - \theta_n)^{-1}).
\]

When \( s' \neq s \), \( \hat{B}_m(x) \) has at most linear convergence for \( \theta_1 \), and

\[
\lim_{x \to \theta_1} \frac{\hat{B}_m(x) - \theta_1}{x - \theta_1} = \frac{s' - s}{s' + s(m-2)}.
\]

**Proof.** When \( s' = s \), equation (3.3) becomes

\[
\hat{h}_k(r_1, \ldots, r_n) = \sum_{i=0}^{k} \hat{h}_i(r_{s'+1}, \ldots, r_n) r_1^{k-i}.
\]
From (3.3), we have
\[
\hat{B}_m(x) - \theta_1 = x - \theta_1 - (x - \theta_1) \frac{(x - \theta_1)^m - 2 h_{m-2}(r_1, \ldots, r_n)}{(x - \theta_1)^m - 1 h_{m-1}(r_1, \ldots, r_n)}
\]
\[
= (x - \theta_1) \left( 1 - \frac{\sum_{i=0}^{m-2} \hat{h}_i(r_{s'+1}, \ldots, r_n)(x - \theta_1)^i}{\sum_{i=0}^{m-1} \hat{h}_i(r_{s'+1}, \ldots, r_n)(x - \theta_1)^i} \right)
\]
\[
= (x - \theta_1)^m \frac{h_{m-1}(r_{s'+1}, \ldots, r_n)}{\sum_{i=0}^{m-1} \hat{h}_i(r_{s'+1}, \ldots, r_n)(x - \theta_1)^i}.
\]

Hence,
\[
\lim_{x \to \theta_1} \frac{\hat{B}_m(x) - \theta_1}{(x - \theta_1)^m} = \lim_{x \to \theta_1} \frac{\hat{h}_{m-1}(r_{s'+1}, \ldots, r_n)}{1 + \sum_{i=0}^{m-1} \hat{h}_i(r_{s'+1}, \ldots, r_n)(x - \theta_1)^i}
\]
\[
= \hat{h}_{m-1}((\theta_1 - \theta_{s'+1})^{-1}, \ldots, (\theta_1 - \theta_{n})^{-1}).
\]

When \(s' \neq s\), by (3.3) and (3.4) we have
\[
\hat{B}_m(x) - \theta_1 = x - \theta_1 - (x - \theta_1) \frac{(x - \theta_1)^m - 2 h_{m-2}(r_1, \ldots, r_n)}{(x - \theta_1)^m - 1 h_{m-1}(r_1, \ldots, r_n)}
\]
\[
= (x - \theta_1) \left( 1 - \frac{(m - 3 + s'/s)_{m-2}/(m - 2)! + \sum_{i=1}^{m-2} a_{m-2,i}(x - \theta_1)^i}{(m - 2 + s'/s)_{m-1}/(m - 1)! + \sum_{i=1}^{m-1} a_{m-1,i}(x - \theta_1)^i} \right).
\]

Hence,
\[
\lim_{x \to \theta_1} \frac{\hat{B}_m(x) - \theta_1}{x - \theta_1} = 1 - \frac{(m - 3 + s'/s)_{m-2}/(m - 2)!}{(m - 2 + s'/s)_{m-1}/(m - 1)!} = \frac{s' - s}{s' + s(m - 2)}.
\]

\[\square\]

**Remark 3.3.** When \(m = 2\) and \(s' \leq s/2\), \(\hat{B}_m(x)\) does not converge to \(\theta_1\) since
\[
\left| \frac{s' - s}{s' + s(m - 2)} \right| \geq 1.
\]

### 3.2. An efficient algorithm for computing \(\hat{B}_m\)

Let \(\hat{Q}(t)\) and \(\hat{H}(t)\) be the generating functions for \(\hat{q}_r\) and \(\hat{h}_r\), respectively. That is,
\[
\hat{Q}(t) = \sum_{r \geq 1} \hat{q}_r t^{r-1} \quad \text{and} \quad \hat{H}(t) = \sum_{r \geq 0} \hat{h}_r t^r.
\]

By (3.1), we have
\[
\hat{Q}(t) = \frac{1}{s} Q(t) = \frac{E'(-t)}{sE(-t)},
\]
and recursion (3.2) implies
\[
\hat{Q}(t) = \frac{\hat{H}'(t)}{\hat{H}(t)}.
\]

Thus,
\[
sE(-t) \hat{H}'(t) = E'(-t) \hat{H}(t),
\]
which gives rise to the following recursion:
\[
(3.6) \quad \hat{h}_i = \frac{1}{s \cdot i} \sum_{r=0}^{i-1} (-1)^{i-r-1}(i - r + s \cdot r) c_{i-r} \hat{h}_r \quad \text{for} \quad i = 1, 2, \ldots.
\]
Given an input $x_0$, an efficient algorithm to compute $\hat{B}_m(x_0)$ is as follows:

**Algorithm 2**: Evaluation of the $m$-th order iteration function $\hat{B}_m$ for roots of multiplicity $s$ at $x_0$.

- **input**: $m, s, p(x)$, and $x_0$.
- **output**: $\hat{B}_m(x_0)$.
- $\hat{h}[0] = 1$;
- for $i$ from 1 to $m-1$ do
  - $c[i] = p^{(i)}(x_0)/(i!p(x_0))$;
- end
- for $i$ from 1 to $m-1$ do
  - $\hat{h}[i] = (\sum_{r=0}^{i-1}(-1)^{i-r-1} (i-r+s \cdot r) c[i-r] \hat{h}[r])/(s \cdot i)$;
- end
- **output** $\hat{B}_m(x_0) = x_0 - \hat{h}[m-2]/\hat{h}[m-1]$;

**Remark 3.4.** For a polynomial $p(x)$ of degree $n$, the normalized derivatives $p^{(i)}(x_0)/i!$ can be evaluated in $O(n \log^2 n)$ time (see Kung [12]). Furthermore, $\hat{h}[m]$, the $m$-th term of a sequence defined via a homogeneous linear recurrence relation, can be computed in $O(n \log n \log m)$ arithmetic operations (see Fiduccia [2]). Therefore, the computational complexity of $\hat{B}_m(x_0)$ is $O(n \log n (\log m + \log n))$.

Using the above algorithm, we can also derive the closed form formulas for $\hat{B}_2, \hat{B}_3$ and $\hat{B}_4$:

- $\hat{B}_2(x) = x - \frac{p(x)}{p'(x)}$;
- $\hat{B}_3(x) = x - 2s \frac{p(x)}{p'(x)}(1 + s) p''(x) - sp(x) p''(x)$;
- $\hat{B}_4(x) = x - 3s \frac{p(x)}{p'(x)}\frac{(1 + s) p'(x)^2 - sp(x) p''(x)}{(2s^2 + 3s + 1)p'(x)^3 - 3s(1+s)p(x)p'(x)p''(x) + s^2 p(x)^2 p'''(x)}$.

**Remark 3.5.** The generalized complete symmetric functions $\hat{h}_i$ also admit an upper Hessenberg determinantal form

$$\hat{h}_i = \begin{vmatrix} c(i, 1)e_1 & c(i, 2)e_2 & \ldots & c(i, i-1)e_{i-1} & c(i, i)e_i \\ 1 & c(i, 1)e_1 & \ddots & \ddots & c(i, i-1)e_{i-1} \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & c(2, 2)e_2 \\ 0 & 0 & \ldots & 1 & c(1, 1)e_1 \end{vmatrix},$$

where coefficients $c(i, j) = (j + s(i - j))/(s \cdot i)$. This follows from the fact that by expanding the determinant along the first row we arrive at recurrence relation [3.6].

When $s = 1$, all the coefficients $c(i, j) = 1$ and the above formula is equivalent to the determinantal representation of Basic Family given by [7].
4. The Connection between $B_m$ and $\hat{B}_m$

In this section, we show that $\hat{B}_m$ is actually the application of $B_m$ to $p(x)^{1/s}$. We start with a generalization of Proposition 2.1.

**Proposition 4.1.** Let $p(x)$ be a polynomial of degree $n$ with complex coefficients, and let $\theta_1, \ldots, \theta_n$ be its complex roots. Let $f(x) = p(x)^{1/s}$.

Define $r_j = 1/(x - \theta_j)$, $j = 1, \ldots, n$. Define

$$\hat{e}_i = \sum_{|\lambda| = i} \frac{m_\lambda}{\lambda!} \prod_{j=1}^{i(\lambda)} \frac{(1/s)\lambda_j}{\lambda_j!},$$

where $m_\lambda$ is the monomial symmetric function corresponding to partition $\lambda = (\lambda_1, \lambda_2, \ldots)$. Then

$$f^{(i)}(x) = \partial f(x) \hat{e}_i(r_1, \ldots, r_n),$$

where $f^{(i)}(x)$ is the $i$-th derivative of $f(x)$.

**Proof.** Write

$$f(x) = c^{1/s} \prod_{i=1}^{n} (x - \theta_i)^{1/s},$$

and apply the generalized product rule of differentiation to it. \qed

The generating function for $\hat{e}_i$ is

$$\hat{E}(t) = \sum_{r \geq 0} \hat{e}_r t^r = \prod_{i \geq 1} (1 + x_i t)^{1/s} = E(t)^{1/s},$$

which is easy to verify using the power series expansion

$$(1 + x_i t)^{1/s} = 1 + \sum_{k=1}^{\infty} \frac{(1/s)_k}{k!} x_i^k t^k.$$

Let $\hat{B}_m$ be the iteration function obtained by applying $B_m$ to $p(x)^{1/s}$. Then by Algorithm 1, we have

$$\hat{B}_m(x) = \frac{\hat{h}_{m-2}(r_1, \ldots, r_n)}{\hat{h}_{m-1}(r_1, \ldots, r_n)}$$

for $m = 2, 3, \ldots$,

where $\hat{h}_0 = 1$, and

$$\hat{h}_i = \sum_{r=0}^{i-1} (-1)^{i-r-1} \hat{e}_{i-r} \hat{h}_r$$

for $i = 1, 2, \ldots$.

Let $\hat{H}(t)$ be the generating function for $\hat{h}_r$. Then from the above recurrence relation between $\hat{h}_r$ and $\hat{e}_r$ we have $\hat{H}(t) = \hat{E}(t)^{-1}$. Thus,

$$\frac{\hat{H}'(t)}{\hat{H}(t)} = \frac{[E(-t)^{-1/s}]'}{E(-t)^{-1/s}} = \frac{E'(-t)}{sE(-t)} = \hat{Q}(t) = \frac{\hat{H}'(t)}{\hat{H}(t)},$$

which implies $\hat{H}(t) = \hat{H}(t)$, and $\hat{h}_r = \hat{h}_r$ for $r = 0, 1, 2, \ldots$.

Therefore, $\hat{B}_m = \hat{B}_m$; i.e., $\hat{B}_m$ is the application of $B_m$ to $p(x)^{1/s}$. 

Remark 4.2. \( \bar{e}_r, r = 0, 1, 2, \ldots \) are generalized elementary symmetric functions, and the recurrence relations among \( \bar{e}_r \), \( \bar{h}_r \) and \( \bar{q}_r \) are the same as those among the standard \( e_r, h_r \) and \( q_r \).

An explicit formula for \( \hat{B}_m \) can be obtained by expanding \( \hat{h}_{m-2} \) and \( \hat{h}_{m-1} \) into functions of \( p(x) \) and its derivatives.

Using the relation between generating functions \( \hat{H}(t) \) and \( E(t) \),

\[
\hat{H}(t) = \sum_{r \geq 0} \hat{h}_r t^r = E(-t)^{-1/s},
\]

we can express \( \hat{h}_r \) as a polynomial in \( e_i \)'s directly.

Define

\[
T = \sum_{r=1}^{\infty} e_r (-t)^r;
\]

then \( E(-t) = 1 + T \), and by Taylor expansion we have

\[
\hat{H}(t) = (1 + T)^{-1/s} = 1 + \sum_{i=1}^{\infty} \frac{(-1/s)^i}{i!} T^i.
\]

Thus,

\[
\hat{h}_k = (-1)^k \sum_{\lambda = (\lambda_1 \ldots \lambda_n)} (-1/s) l(\lambda) \prod_{j=1}^{n} a_j! e_{\lambda_j} \prod_{j=1}^{n} a_j! \lambda_j! \prod_{j=1}^{n} [p(x)_{\lambda_j}]^{a_j}.
\]

REFERENCES


QUANTITATIVE RESEARCH, INTEREST RATE DERIVATIVES, J.P. MORGAN, 270 PARK AVENUE, NEW YORK, NEW YORK 10017
E-mail address: yi.x.jin@jpmorgan.com

DEPARTMENT OF COMPUTER SCIENCE, RUTGERS UNIVERSITY (NEW BRUNSWICK), HILL CENTER-BUSCH CAMPUS, 110 FRELINGHUYSEN ROAD, PISCATAWAY, NEW JERSEY 08854
E-mail address: kalantar@cs.rutgers.edu