다항식의 분해

성준경[†] Gershon Elber[‡] 김명수[†]

† 서울대학교 컴퓨터공학부

서울대학교 컴퓨터연구소

‡ Technion-Israel Institute of Technology

Polynomial Decomposition and Its Applications

Joon-Kyung Seong[†] Gershon Elber[‡] Myung-Soo Kim[†]

† School of Computer Science end Engineering, SNU

Institute of Computer Technology, SNU

‡ Technion-Israel Institute of Technology

요 약

본 논문에서는 다항식 $H(x)$로부터 $H(x) = f(g(x))$를 만족하는 두 개의 다항식 $f(x), g(x)$ 로 분해하는 알고리즘을 제시한다. 간단한 구조의 비선형 방정식 들을 풀면 이 문제를 해결할 수 있다. 일반적으로 비선형 방정식을 푸는 문제는 상당히 어렵다고 여겨지고 있지만, 이 문제에서는 변수들 간의 점화식을 이용하여 정확하게 해를 구할 수 있다. 논문에서는 몇 가지 예제를 통해서 제시한 알고리즘의 효율성을 보이고, 기하 모델링 분야에서 중요한 응용에 대해 논하고자 한다.

1. 서론

두 다항식의 합성은 기하 모델링이나 컴퓨터 그래픽스 분야 등에서 매우 중요하다. Maple이나 Mathematica와 같은 소프트웨어 페키지는 내부 연산자로서 다항식의 합성을 지원하고 있다. 이 논문에서는 합성된 다항식, $H(x) = f(g(x))$, 에 대해서 합성 연산의 역 연산인 decomposition 을 생각하고자 한다. 즉, $H(x)$가 주어졌을 때, $f(x)$ 와 $g(x)$를 구하고자 한다.

이 연산은 Maple이나 Mathematica에서도 아직 지원되지 않고 있다. 일반적으로 합성 연산이 아주 많이 사용되고 있음에도 불구하고, 그 역 연산에 대한 지원이 없다는 것은 놀라운 일이 아닐 수 없다. 역 연산이 없다는 사실로부터 스플라인 모델의 watermarking에 있어서 다항식의 합성이 사용될 수 있다는 주장들이 제기되기도 하였다 [5]. 따라서, 본 연구의 결과는 다항식의 합성을 사용해서는 더 이상 안전하게 모델을 watermarking할 수 없다는 것을 보여주기도 한다.

합성 연산의 역 연산은 같은 곡선이지만 다르게 표현된 두 곡선을 찾아내는 데도 중요한 역할을 한다. NURBS 곡선을 사용할 때에는 degree raising, refinement 또는 composition을 통해서 두 곡선이 다르게 표현 될 수 있다. 하지만, degree raising이나 refinement에 대한 역 연산은 쉽게 구현할 수 있으며, 이 논문에서는 함수의 합성에 관한 역 연산을 다루고자 한다. 곡면이 여러 개의 patche들로 이루어져 있을 때, 각 patch의 경계곡선을 생각해보자. 이들 경계곡선은 같은 모양의 곡선이지만 degree등이 달라져서, tessellation과정에서 hole을 만들기도 한다. 이들 곡선들이 하나의 기본 형식으로 표현 될 수 있다면, matching 과정이 매우 쉬워질 수 있다.

곡선의 합성은 degree raising이나 refinement 과정과 마찬가지로 곡선의 parameterization에도 영향을 주게 된다. 다항식 $c(t) = t[(1-t)+t]^n$ 를 사용해서 합성한 곡선 $f(c(t))$는 곡선의 속도는 보존하지만 차수는 높이게 된다. 하지만 그 이외의 다항식 $c(t)$는 곡선의 속도 또한 변화시키게 된다. Degree raising이나 refinement를 통해서 곡선의 차수가 높아진 경우에는 degree reduction알고리즘으로 차수를 낮출 수 있으며 [6], knot removal을 통해서 중복되는 knot을 제거할 수 있다 [4]. 본 논문에서는 다항식 합성의 역 연산을 제공함으로써 곡선의 차수를 낮추는 새로운 결과를 제시하고자 한다.

두 비선형 다항식 $f(x)$ 와 $g(x)$를 생각해보자. 두 다항식의 합성 $(f \circ g)(x) = f(g(x))$ 은 차수가 훨씬 높고, 많은 비선형 계수를 가지는 다항식이 된다. 일반적으로, 연립 비선

형식을 푸는 문제는 매우 어렵다. 하지만, 본 논문에서 다루고 있는 문제에 있어서 비선형 식들은 매우 정형화된 형태를 가지고 있으며, 이를 이용하면 비교적 쉽게 해를 구할 수 있다. $f(x)$가 유리함수인 경우에도 이러한 정형화된 형태는 projective 공간에서 살펴보면 동일하다.

이 연구의 주요 결과는 다음과 같다: 합성된 다항식 $H(x)$의 계수 중 낮은 차수의 계수는 $f(x)$ 를 구하는데 사용되고, 높은 차수의 계수는 $g(x)$를 구하는데 사용된다. $H(x)$의 계수는 차수가 매우 높을 수 있다. 하지만, convolution과 마찬가지로 차수가 가장 낮은 쪽이나 가장 높은 쪽은 가장 간단한 구조를 가지게 된다. 일반적인 합성 $H(x) = f(g(x))$를 다루는데 있어서 $f(x)$ 와 $g(x)$는 다음과 같은 canonical 형식을 가정해도 일반성을 잃지 않음을 2장에서 보이도록 한다:

$$
\begin{aligned}
f(x) &= x^m + b_{m-1}x^{m-1} + \cdots + b_1 x + b_0, \\
g(x) &= x^k + c_{k-1}x^{k-1} + \cdots + c_1 x.
\end{aligned} \tag{1}
$$

이렇게 가정함으로써, 합성된 다항식의 계수를 상당히 간단하게 만들 수 있으며, 그 구조를 분석하는데도 훨씬 쉬워진다.

식 (1)의 두 다항식을 합성하게 되면 차수가 km인 다항식을 얻게 된다:

$$
\begin{aligned}
H(x) &= x^{km} + a_{km-1}x^{km-1} + \cdots a_1 x + a_0 \\
&= f(g(x)) \\
&= (x^k + c_{k-1}x^{k-1} + \cdots + c_1 x)^m \\
&\quad + b_{m-1}(x^k + c_{k-1}x^{k-1} + \cdots + c_1 x)^{m-1} \\
&\quad \cdots \\
&\quad + b_1(x^k + c_{k-1}x^{k-1} + \cdots + c_1 x) \\
&\quad + b_0.
\end{aligned}
$$

$H(x)$의 최고차항 k개를 살펴보면, 각 계수 c_{k-i}는 a_{km-i}와 $c_k, \cdots, c_{k-i+1}$ ($i = 1, \cdots, k-1$; $c_k = 1$)로 주어지는 함수로서 표현되어진다는 사실을 알 수 있다. 일단 $g(x)$의 계수가 이렇게 구해지고 나면, $f(x)$의 계수와 a_i와의 관계는 선형이다. 따라서, $f(x)$의 계수 b_j는 a_j, $c_1, \cdots, c_{k-1}$, 와 $b_0, \cdots, b_{j-1}$, for $j = 0, \cdots, m-1$ 의 함수로서 정확하게 구해진다.

이 논문은 다음과 같이 구성되어있다. 제 2절에서는 $f(x)$와 $g(x)$를 식 (1) 에서와 같이 간단하게 가정할 수 있음을 보인다. 이를 바탕으로 제 3절에서는 주어진 곡선을 분해하는 알고리즘에 대해 기술한다. 마지막으로, 제 4절에서는 논문에 대한 결론을 내린다.

2. A Template Solution

이 절에서는 다항식 $H(x) = f(g(x))$를 두 개의 다항식 $f(x)$와 $g(x)$로 분해하는 과정을 간단하게 만들어주는 affine 변환에 대해서 기술하고자 한다.

선형 다항식에 의해 표현되는 affine 변환을 생각해보자:

$$l(x) = \alpha x + \beta,$$

여기서 $\alpha \neq 0$ 이고 β는 상수이다. 그러면, 그 역 변환 역시 선형 다항식이 된다:

$$l^{-1}(x) = \alpha^{-1}(x - \beta).$$

합성된 다항식 $H(x) = f(g(x))$가 주어질 때, 위의 선형 다항식을 사용하면 $H(x)$를 분해 하는 방법은 무수히 많다:

$$H(x) = f(g(x)) = \hat{f}(\hat{g}(x)),$$

단,

$$
\begin{aligned}
\hat{f}(x) &= (f \circ l^{-1})(x), \\
\hat{g}(x) &= (l \circ g)(x).
\end{aligned}
$$

만약, $f(x)$ 와 $g(x)$가 다항식이라면 $\hat{f}(x) = (f \circ l^{-1})(x)$ 와 $\hat{g}(x) = (l \circ g)(x)$ 또한 차수가 같은 다항식이된다.

$g(x)$가 $c_k \neq 0$를 만족하는 차수 k의 다항식이라고 가정하자:

$$g(x) = c_k x^k + c_{k-1}x^{k-1} + \cdots + c_1 x + c_0,$$

이 때, 다음과 같은 선형 다항식을 사용하면,

$$l(x) = c_k^{-1}(x - c_0),$$

$l(g(x))$는 다음과 같다:

$$\hat{g}(x) = x^k + (c_k^{-1}c_{k-1})x^{k-1} + \cdots + (c_k^{-1}c_1)x.$$

따라서, 일반성을 잃지 않고 $c_k = 1$ 이고 $c_0 = 0$이라고 가정할 수 있다.

이제 다항식 $f(x)$가 $b_m \neq 0$을 만족하는 차수 m인 다항식이고, $g(x)$가 위와 같이 주어졌다고 가정하면:

$$
\begin{aligned}
f(x) &= b_m x^m + b_{m-1}x^{m-1} + \cdots + b_1 x + b_0, \\
g(x) &= x^k + c_{k-1}x^{k-1} + \cdots + c_1 x,
\end{aligned}
$$

그 합성 다항식 $H(x) = f(g(x))$는 다음과 같다:

$$
\begin{aligned}
H(x) &= a_{km}x^{km} + a_{km-1}x^{km-1} + \cdots a_1 x + a_0 \\
&= f(g(x))
\end{aligned}
$$

$$
\begin{aligned}
&= b_m(x^k + c_{k-1}x^{k-1} + \cdots + c_1x)^m \\
&\quad + b_{m-1}(x^k + c_{k-1}x^{k-1} + \cdots + c_1x)^{m-1} \\
&\quad \cdots \\
&\quad + b_1(x^k + c_{k-1}x^{k-1} + \cdots + c_1x) \\
&\quad + b_0.
\end{aligned}
$$

이때, 차수가 km인 항의 계수를 비교해보면, $a_{km} = b_m$인 것을 알 수 있다. 위의 식에서 $H(x)$와 $f(x)$를 $a_{km} = b_m$로 나누면, 두 다항식 모두 최고차항의 계수가 1인 다항식으로 가정할 수 있다. 따라서, 이 논문에서는 다음과 같이 가정한다:

$$
\begin{aligned}
H(x) &= x^{km} + a_{km-1}x^{km-1} + \cdots a_1x + a_0, \\
f(x) &= x^m + b_{m-1}x^{m-1} + \cdots + b_1x + b_0, \\
g(x) &= x^k + c_{k-1}x^{k-1} + \cdots + c_1x
\end{aligned}
$$

따라서, 합성된 식은 다음과 같다:

$$
\begin{aligned}
f(g(x)) &= (x^k + c_{k-1}x^{k-1} + \cdots + c_1x)^m \qquad (2) \\
&\quad + b_{m-1}(x^k + c_{k-1}x^{k-1} + \cdots + c_1x)^{m-1} \\
&\quad \cdots \\
&\quad + b_1(x^k + c_{k-1}x^{k-1} + \cdots + c_1x) \\
&\quad + b_0.
\end{aligned}
$$

3. Recurrence Formulas

이 절에서는 c_{k-i} 와 b_j를 점진적으로 구할 수 있는 점화 관계식에 대해 기술한다.

먼저, 변수 $\mathcal{A}[l,i]$, $(l = 1, \cdots, m;\ i = 0, \cdots, kl - l)$를 다음과 같이 정의하자.

$$(c_kx^k + c_{k-1}x^{k-1} + \cdots + c_1x)^l = \sum_{i=0}^{kl-l} \mathcal{A}[l,i]x^{kl-i},$$

단, $c_k = 1$ 이다. 그러면,

$$
\begin{aligned}
\mathcal{A}[1,i] &= c_{k-i}, \quad \text{for } i = 0, \cdots, k-1; \\
\mathcal{A}[l,0] &= c_k^l = 1, \quad \text{for } l = 1, \cdots, m; \text{ and} \\
\mathcal{A}[m,i] &= a_{km-i}, \quad \text{for } i = 1, \cdots, k-1.
\end{aligned}
$$

임을 알 수 있다.

또한, $l = 2, \cdots, m$에 대해서 다음의 점화식을 사용하면,

$$
\begin{aligned}
&(x^k + c_{k-1}x^{k-1} + \cdots + c_1x)^l \\
&= (x^k + c_{k-1}x^{k-1} + \cdots + c_1x) \\
&\quad \cdot (x^k + c_{k-1}x^{k-1} + \cdots + c_1x)^{(l-1)} \\
&= (x^k + c_{k-1}x^{k-1} + \cdots + c_1x) \\
&\quad \cdot \left(\sum_{i=0}^{k(l-1)-(l-1)} \mathcal{A}[l-1,i]x^{k(l-1)-i} \right),
\end{aligned}
$$

$i = 0, \cdots, k-1$에 대해서 다음과 같은 점화 관계식을 유도할 수 있다:

$$\mathcal{A}[l,i] = \sum_{j=0}^{i} c_{k-i+j}\mathcal{A}[l-1,j].$$

그러면, 수학적 귀납법에 의해서 $\mathcal{A}[l,i]$는 $c_k, \cdots, c_{k-i}$로부터 결정된다는 사실을 쉽게 보일 수 있다.

이 점화 관계식을 사용해서 다음을 이끌어낼 수 있다:

$$
\begin{aligned}
a_{km-i} &= \mathcal{A}[m,i] \\
&= \sum_{j=0}^{i} c_{k-i+j}\ \mathcal{A}[m-1,j] \\
&= c_{k-i}\mathcal{A}[m-1,0] + \mathcal{A}[m-1,i] + \sum_{j=1}^{i-1} c_{k-i+j}\ \mathcal{A}[m-1,j] \\
&= c_{k-i} + c_{k-i}\mathcal{A}[m-2,0] + \mathcal{A}[m-2,i] \\
&\quad + \sum_{j=1}^{i-1} c_{k-i+j}\ \mathcal{A}[m-2,j] + \sum_{j=1}^{i-1} c_{k-i+j}\ \mathcal{A}[m-1,j] \\
&= 2c_{k-i} + \mathcal{A}[m-2,i] + \sum_{l=m-2}^{m-1}\sum_{j=1}^{i-1} c_{k-i+j}\ \mathcal{A}[l,j] \\
&= (m-1)c_{k-i} + \mathcal{A}[1,i] + \sum_{l=1}^{m-1}\sum_{j=1}^{i-1} c_{k-i+j}\ \mathcal{A}[l,j] \\
&= mc_{k-i} + \sum_{l=1}^{m-1}\sum_{j=1}^{i-1} c_{k-i+j}\ \mathcal{A}[l,j]
\end{aligned}
$$

따라서, $i = 1, 2, \cdots, k-1$에 대해 우리는 다음의 결론을 유도해 낸다:

$$c_{k-i} = \frac{a_{km-i} - \sum_{l=1}^{m-1}\sum_{j=1}^{i-1} c_{k-i+j}\ \mathcal{A}[l,j]}{m}. \qquad (3)$$

각 항목 $\mathcal{A}[l,j]$가 $c_k, \cdots, c_{k-j}$에 의해 결정된다는 것을 주지하면, c_{k-i}가 a_{km-i} 와 $c_k, \cdots, c_{k-i+1}$에 의해서 계산되어진다는 것을 알 수 있다.

일단 $c_{k-1}, \cdots, c_1$가 위의 점화 관계식을 사용해서 구해지고 나면, 다항식 $f(x)$의 계수 b_i는 $c_{k-1}, \cdots, c_1$ 와 $b_0, \cdots, b_{i-1}$의 rational 표현으로 나타내어 질 수 있다. 이를 위해서 새로운 변수 $\mathcal{B}[l,i]$, $(l = 1, \cdots, m;\ i = l, \cdots, kl)$를 다음과 같이 정의하자:

$$(x^k + c_{k-1}x^{k-1} + \cdots + c_1x)^l = \sum_{i=l}^{kl} \mathcal{B}[l,i]x^i.$$

이제 다음의 식이 성립한다:

$$\begin{aligned}\mathcal{B}[1,i] &= c_i, \quad \text{for } i=1,\cdots,k-1; \text{and} \\ \mathcal{B}[l,l] &= c_1^l, \quad \text{for } l=1,\cdots,m.\end{aligned}$$

$\mathcal{B}[l,i]$의 각 계수 값은 다음과 같이 구할 수 있다.

$$\begin{aligned}\sum_{i=l}^{kl}\mathcal{B}[l,i]x^i &= (x^k+c_{k-1}x^{k-1}+\cdots+c_1x)^l \\ &= (x^k+c_{k-1}x^{k-1}+\cdots+c_1x) \\ &\quad \cdot(x^k+c_{k-1}x^{k-1}+\cdots+c_1x)^{(l-1)} \\ &= (x^k+c_{k-1}x^{k-1}+\cdots+c_1x) \\ &\quad \cdot\left(\sum_{i=l-1}^{k(l-1)}\mathcal{B}[l-1,i]x^i\right).\end{aligned}$$

그러면, $B[l,i]$를 위한 점화식은 다음과 같다.

$$\mathcal{B}[l,i]=\sum_{j=1}^{k}c_j\mathcal{B}[l-1,i-j].$$

여기서 $i-j<l-1$ 이고 $i-j>k(l-1)$일 때, $\mathcal{B}[l-1,i-j]=0$ 이기 때문에 다음을 얻을 수 있다:

$$\mathcal{B}[l,i]=\sum_{j=\max(1,i-k(l-1))}^{\min(k,i-l+1)}c_j\mathcal{B}[l-1,i-j].$$

이렇게하여 $\mathcal{B}[l,i]$의 값을 모두 계산하고 나면, b_l, $(l=1,\cdots,m-1)$을 계산할 수 있다. 먼저, $l<p$ 또는 $pk<l$ 일 때, $\mathcal{B}[p,l]=0$ 임을 가정하면, 식 (2) 으로부터

$$a_l=b_lc_1^l+\sum_{p=1}^{l-1}b_p\mathcal{B}[p,l].$$

를 얻을 수 있다. 따라서, $c_1\neq 0$일 때, b_l은 다음과 같이 구할 수 있다:

$$b_l=\frac{a_l-\sum_{p=1}^{l-1}b_p\mathcal{B}[p,l]}{c_1^l}.$$

이제 다음과 같이 일반적인 경우를 생각해보자.

$$g(x)=x^k+c_{k-1}x^{k-1}+\cdots+c_sx^s;$$

그러면

$$a_{sl}=b_lc_s^l+\sum_{p=1}^{l-1}b_p\mathcal{B}[p,sl],$$

이고, 최종적으로 다음을 얻는다.

$$b_l=\frac{a_{sl}-\sum_{p=1}^{l-1}b_p\mathcal{B}[p,sl]}{c_s^l}. \tag{4}$$

다음의 알고리즘 1, 2는 $f(x)$ 와 $g(x)$의 계수를 구하는 방법을 보여준다. n의 모든 devisor k $(1<k<n)$에 대해서, $m=n/k$이라 두고 알고리즘 1, 2를 실행하면, $f(x)$ 와 $g(x)$를 구할 수 있고, $H(x)=f(g(x))$가 성립하는지 검증한다. $n<30$인 경우 두 알고리즘은 실시간에 실행된다.

계산상의 수치적 에러로 인해 $H(x)$ 와 $f(g(x))$의 계수가 약간 다를 수 있다. 논문에서는 두 다항식의 계수 차이가 10^{-9}보다 작을 경우 같은 것으로 간주하였다.

4. Conclusion

이 논문에서는 합성된 다항식 $H(x)$가 주어졌을 때 $H(x)=f(g(x))$를 만족하는 두 다항식 $f(x)$ 와 $g(x)$를 정확하게 구해내는 알고리즘을 기술하였다.

이 결과를 이용해서 우리는 주어진 polynomial/rational 곡선이 재매개화된 곡선인지의 여부를 쉽게 알 수 있었다. 원래의 스플라인 곡선과 재매개화 함수까지도 정확하게 구해낼 수 있었다.

이 논문에 제시된 알고리즘은 자유 곡선 모델링 분야에서 많은 흥미로운 문제들에 응용될 수 있을 것이다. 앞으로 해결되어야 할 중요한 문제로는 2변수로 된 다항식의 decomposition 문제를 들 수 있다.

참고 문헌

[1] Cohen, E., Riesenfeld, R.F., and Elber, G. *Geometric Modeling with Splines: An Introduction*. A.K. Peters, Natick Massachusetts, 2001.

[2] Heal, K.M., Hansen, M., and Richard, K. *Maple V Learning Guide* (for Release 5), Springer-Verlag, 1997.

[3] IRIT 8.0 User's Manual, October 2000, Technion. http://www.cs.technion.ac.il/~irit.

[4] Lyche, T., and Morken, V. Knot removal for parametric B-spline curves and surfaces *Computer Aided Geometric Design*, Vol. 4, No. 3, pp. 217–230, 1987.

[5] Ohbuchi, R., Masuda, H., and Aono, M. A Shape-preserving Data Embedding Algorithm for NURBS Curves and Surfaces. *Proc. of Computer Graphics International (CGI 99)*, Canada, June 1999.

[6] Piegl, L., and Tiller, W. Algorithm for degree reduction of B-spline curves. *Computer-Aided Design*, Vol. 27, No. 2, pp. 101–110, 1995.

Algorithm 1

Input: The composite function $\mathcal{H}(x)$ of degree n;

Output: The function $g(x)$ of degree k which satisfies that $f(g(x)) = \mathcal{H}(x)$;

DecompositionG($\mathcal{H}(x)$)

```
begin
    for i := 1 to k − 1 do
        A[m, i] ⇐ a_{km−i};
    end
    for l := 1 to m do
        A[l, 0] ⇐ 1;
    end
    for i := 1 to k − 1 do
        c_{k−i} ⇐ (1/m) ( a_{km−i} − Σ_{l=1}^{m−1} Σ_{j=1}^{i−1} c_{k−i+j} A[l, j] ); /% Equation (3) %/
        for l := m − 1 downto 1 do
            A[l, i] ⇐ A[l + 1, i] − Σ_{j=0}^{i−1} c_{k−i+j} A[l, j];
        end
    end
    c_0 = 0;
    Construct a polynomial function g(x) using c_i, i = 0, 1, ···, k;
    return g(x);
```

where, in mathematical notation:

$c_{k-i} \Leftarrow \frac{1}{m}\left(a_{km-i} - \sum_{l=1}^{m-1}\sum_{j=1}^{i-1} c_{k-i+j}\ \mathcal{A}[l,\ j]\right)$; /% Equation (3) %/

$\mathcal{A}[l,\ i] \Leftarrow \mathcal{A}[l+1,\ i]\ -\ \sum_{j=0}^{i-1} c_{k-i+j}\ \mathcal{A}[l,\ j]$;

Algorithm 2

Input: The composite function $\mathcal{H}(x)$ of degree n;
The function $g(x)$ of degree k;

Output: The function $f(x)$ of degree m which satisfies that $f(g(x)) = \mathcal{H}(x)$;

DecompositionH($\mathcal{H}(x),\ g(x)$)

```
begin
    /% Assume g(x) = x^k + c_{k−1} x^{k−1} + ··· + c_s x^s %/
    for i := 1 to k − 1 do
        B[1, i] ⇐ c_i;
    end

    for l := 2 to m do
        for i := 1 to kl do
            B[l, i] ⇐ Σ_{j=max(1, i−k(l−1))}^{min(k, i−l+1)} c_j B[l − 1, i − j];
        end
    end
    for l := 1 to m − 1 do
        b_l ⇐ (1/c_s^l) ( a_{sl} − Σ_{p=1}^{l−1} b_p B[p, sl] ); /% Equation (4) %/
    end
    b_0 ⇐ a_0;
    Construct a polynomial function f(x) using b_l, l = 0, 1, ···, m − 1;
    return f(x);
```

where, in mathematical notation:

$g(x) = x^k + c_{k-1}x^{k-1} + \cdots + c_s x^s$

$\mathcal{B}[l,i] \Leftarrow \sum_{j=\max(1,i-k(l-1))}^{\min(k,i-l+1)} c_j \mathcal{B}[l-1,i-j]$;

$b_l \Leftarrow \frac{1}{c_s^l}\left(a_{sl}\ -\ \sum_{p=1}^{l-1} b_p \mathcal{B}[p,sl]\right)$; /% Equation (4) %/

[7] Spiegel, M. *Mathematical Handbook of Formulas and Tables.* Schaum's Outline Series in Mathematics, McGraw-Hill, pp 4, 1968.

[8] Wolfram, S. *The Mathematica Book*, 3rd ed. Wolfram Media/Cambridge University Press, 1996.