

Poszukiwanie węzłów idealnych II – algorytm wyznaczania wielomianów Alexandra w języku Maple V.4¹⁾

Jacek BUŚKIEWICZ[†], Piotr PIERAŃSKI[‡]

Politechnika Poznańska

[†]Wydział Budowy Maszyn, [‡]Wydział Fizyki Technicznej
ul. Piotrowo 3, 60-965 Poznań

e-mail: pieransk@phys.put.poznan.pl

Otrzymano 28 listopada 1997 roku

Streszczenie. W artykule przedstawiono algorytm wyznaczania wielomianów Alexandra zaimplementowany w języku obliczeń symbolicznych Maple oraz znaleziono te wielomiany dla wszystkich węzłów prostych o 9 skrzyżowaniach.

Słowa kluczowe: węzeł, wielomian Alexandra, Maple

1. Wprowadzenie

Poszukiwanie węzłów idealnych [1, 2], a więc tych konfiguracji węzłów wykonanych z wiotkiej, ale twardej rurki, dla których iloraz długości rurki i jej średnicy osiąga globalne minimum, nie jest zadaniem prostym. Wszystkie algorytmy poszukujące węzłów idealnych muszą spełniać podstawowy warunek: generowane przez te algorytmy zmiany konfiguracji węzła nie mogą zmieniać jego typu. Pojawia się więc problem stwierdzenia, czy konfiguracja końcowa, do której dany algorytm zdołał dotrzeć, to węzeł tego samego typu, co konfiguracja wyjściowa. Identyfikacji typu węzła (niestety niekiedy niejednoznacznej) można dokonać wyznaczając jego wielomian Alexandra. Przypuśćmy, że dane są dwa węzły (przez węzeł rozumiemy będziemy zamkniętą krzywą w przestrzeni R^3) mające na pierwszy rzut oka zupełnie inny kształt. Odpowiedź na pytanie, czy są to jednak węzły tego samego typu, tzn. czy można jeden z nich przekształcić w drugi na drodze ciągłej deformacji (ściskanie, skręcanie itd.), podczas której nie dochodzi do samoprzecięcia się krzywej, nie jest prosta. Jedną z możliwości jest wykonanie modeli obu węzłów, np. ze sznurka, i poszukiwanie odpowiedzi na drodze manualnej. Przy skomplikowanej postaci węzłów, rozwiązanie tego problemu może okazać się bardzo pracochłonne. Inna możliwość, to topologiczna analiza obu węzłów prowadząca do wyznaczenia związanych z nimi niezmienniczych obiektów algebraicznych.

¹⁾ Praca wykonana w ramach grantu 8T11 F010 08p04.

2. Wielomian Alexandra

W 1928 roku J. W. Alexander pierwszy podał reguły pozwalające przypisać każdemu węzłowi pewien wielomian. Wielomian Alexandra jest niezmiennikiem węzła: nie ulega zmianie, jeśli węzeł poddamy deformacjom, podczas których nie dochodzi do samoprzecięcia. Sytuacje, w których dwa różne węzły związane są z tym samym wielomianem, okazały się rzadkie; na tyle rzadkie, by wyznaczanie wielomianu Alexandra stało się użyteczną metodą identyfikacji węzłów.

Każdy węzeł można sprowadzić do postaci, która charakteryzuje się minimalną liczbą skrzyżowań MCN (skr. ang. *minimal crossing number*) widocznych w jego rzucie na pewną płaszczyznę [4]. I tak np. mamy jeden węzeł, którego $MCN = 0$ (węzeł trywialny). Podobnie, istnieje jeden węzeł o $MCN = 3$ i jeden węzeł o $MCN = 4$. Istnieją po dwa węzły o $MCN = 5$ i $MCN = 6$. Wraz ze wzrostem minimalnej liczby skrzyżowań bardzo gwałtownie wzrasta liczba różnych węzłów. Istnieją tablice węzłów, w których węzły poklasyfikowano według charakteryzującej je minimalnej liczby skrzyżowań. W tablicach tych podaje się też często wielomiany Alexandra sklasyfikowanych tam węzłów. Znajomość wielomianu Alexandra węzła, nad którym pracujemy, pomaga odnaleźć go w tablicach i określić typ.

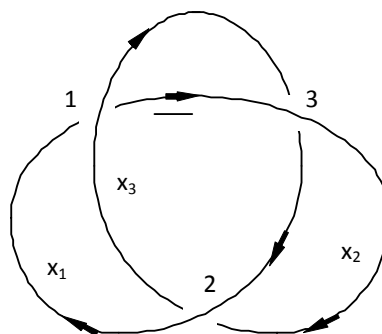
3. Algorytm wyznaczania wielomianu Alexandra

Przedstawiony poniżej algorytm wyznaczania wielomianu Alexandra opisany został przez Vologodskiego [4].

Rzutowujemy węzeł na dowolnie wybraną płaszczyznę i zaznaczamy na otrzymanym rzucie, które z krzyżujących się segmentów bieżą góra, a które dołem. Rzut ten nazwijmy projekcją. Zakładamy, że żadne dwa skrzyżowania nie znajdują się w tym samym punkcie płaszczyzny rzutowania.

Numerowanie skrzyżowań

Zorientujmy projekcję, tzn. wybierzmy i zaznaczmy na rzucie jeden z dwu możliwych kierunków poruszania się wzdłuż projekcji węzła. Następnie, zaczynając od dowolnego punktu węzła nie leżącego na skrzyżowaniu, poruszajmy się wzdłuż krzywej projekcji zgodnie z obranym kierunkiem. Napotykając podczas tej podróży skrzyżowanie, sprawdzamy, czy wjeżdżamy na nie drogą biegnącą dołem czy góra. Jeśli droga, którą się aktualnie poruszamy, biegnie dołem, przejeżdżane skrzyżowanie numerujemy kolejną liczbą naturalną k . Powrót do punktu wyjścia oznacza, że wszystkie skrzyżowania zostały przejechane dwukrotnie: raz góra, raz dołem. Wszystkie zostały więc ponumerowane. Niech będzie ich n .



Rys. 1. Projekcja węzła z obranym kierunkiem, skrzyżowaniami 1, 2 i 3 oraz generatorami x_1 , x_2 i x_3

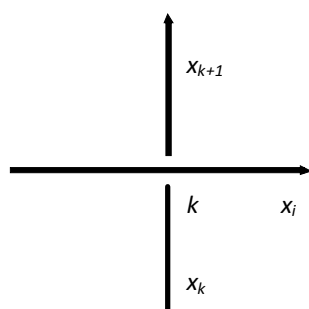
Określenie typu skrzyżowań

Orientacja krzywej projekcji określona przez wybrany kierunek ruchu pozwala na określenie parametru skrzyżowania zwanego jego typem. Skrzyżowanie jest typu 1, jeśli przejeżdżając skrzyżowanie drogą biegnącą górą, widzimy, że ruch na drodze dolnej odbywa się w lewo. Jeśli ruch na drodze dolnej odbywa się w prawo, skrzyżowanie jest typu 2. Oznaczmy typ k -tego skrzyżowania przez T_k .

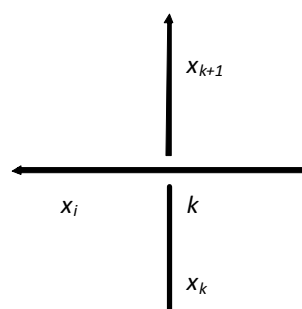
Wyznaczenie generatorów

Podzielmy krzywą projekcji węzła na zbiór krzywych zwanych generatorami: k -ty generator to część krzywej projekcji między $(k - 1)$ -szym i k -tym skrzyżowaniem. Ponieważ skrzyżowania numerowane były podczas przejeżdżania przez nie drogą biegnącą dołem, przez wszystkie znajdujące się na nim skrzyżowania generator przechodzi górą. Przyjmijmy, że przechodząc górą przez skrzyżowanie generator zostawia na nim swój numer G_k .

Każde skrzyżowanie jest więc scharakteryzowane przez trzy parametry: numer skrzyżowania k , typ T_k i numer G_k przechodzącego przez nie generatora.



Rys. 2a



Rys. 2b

Na rys. 2a i 2b przedstawiono skrzyżowanie o numerze k i numerze generatora $G_k = i$ przechodzącego przez nie, przy czym na rys. 2a skrzyżowanie jest typu 1, a na rys. 2b – typu 2.

Określenie macierzy Alexandra

Znajomość wszystkich wartości T_k i G_k ($k = 1, 2, \dots, n$) pozwala na skonstruowanie macierzy kwadratowej A stopnia n , zwanej macierzą Alexandra, w której k -ty wiersz związany jest z k -tym skrzyżowaniem. Dalej podajemy reguły konstrukcji macierzy Alexandra.

1. Wszystkie elementy macierzy, oprócz a_{kk} , $a_{k, k+1}$ i a_{ki} , gdzie $i = G_k$, są zerami. Niezerowe elementy wyznacza się w następujący sposób:
2. Jeśli $i = k$ lub $i = k + 1$, to niezależnie od typu skrzyżowania mamy

$$a_{kk} = -1, \quad a_{k,k+1} = 1.$$

3. Jeśli $i \neq k$ oraz $i \neq k + 1$, to

- dla pierwszego typu skrzyżowania

$$a_{kk} = 1, \quad a_{k,k+1} = -t, \quad a_{ki} = t - 1,$$

- dla drugiego typu skrzyżowania

$$a_{kk} = -t, \quad a_{k,k+1} = 1, \quad a_{ki} = t - 1.$$

Reguły te obowiązują dla $k = 1, 2, \dots, n$, przy czym $n + 1$ zastępujemy przez 1.

Wyznaczenie wielomianu Alexandra

Wielomian Alexandra $\Delta(t)$ uzyskujemy z macierzy Alexandra A , obliczając dowolny jej minor rzędu $n - 1$ (wynik nie zależy od wyboru minoru) i mnożąc wynik przez $\pm t^m$ (m oznacza liczbę całkowitą) tak, by wielomian nie miał ujemnych potęg i jego wyraz wolny był dodatni.

Widzimy zatem, że problem równoważności węzłów sprowadza się do porównywania wielomianów skończonego rzędu, co nie stanowi żadnej trudności.

Wielomian Alexandra nie jest kompletnym niezmiennikiem węzłów. Zamiana wszystkich typów skrzyżowań z 1 na 2 i odwrotnie nie zmienia wielomianu Alexandra. Operacja ta równoważna jest, jak łatwo się przekonać, przejściu od danego węzła do jego zwierciadlanego odbicia. Wielomian Alexandra nie pozwala zatem odróżnić węzła od jego zwierciadlanego odbicia. W przypadku węzłów achiralnych, które dają się przeformować bez samoprzecięć w swe zwierciadlane odbicia, jest to naturalne, ale w przypadku węzłów chiralnych, świadczy o słabości wielomianu Alexandra jako indykatora typu węzła.

Zauważmy, że zmiana orientacji projekcji węzła (a więc i orientacji samego węzła) nie zmienia otrzymanego z jej analizy wielomianu Alexandra. Dlatego wielomian Alexandra charakteryzuje jednoznacznie węzeł niezorientowany. Ponadto, jak już wspomnieliśmy, może zdarzyć się, że dwa różne węzły mają ten sam wielomian Alexandra. Wśród 84 niezorientowanych węzłów o liczbie skrzyżowań od 0 do 9, pięć par węzłów nie jest rozróżnialnych przez wielomian Alexandra.

4. Implementacja algorytmu wyznaczającego wielomian Alexandra w języku Maple V.4

Przyjmijmy, że z analizy projekcji pewnego węzła otrzymaliśmy wszystkie parametry charakteryzujące jej n skrzyżowań, tzn. wektory G i T . Odpowiedni program zaimplementowany w pakiecie Maple, realizujący opisany algorytm wyznaczania wielomianu Alexandra, ma następującą postać:

```
>with(linalg):
# n - liczba punktów
# T - wektor typów
# G - wektor generatorów
```

```

> wielomian:=proc(n,G,T)
>   local A,i,k,w,B,f,st,wsp;
>   A:=matrix(n,n);
>   for i to n do
>     for k to n do
>       A[i,k]:=0;
>     od;
>   od;
>   for i to n do
>     if i=n then w:=1 else w:=i+1 fi;
>     if (G[i]=i) or (G[i]=w) then
>       A[i,i]:=-1;
>       A[i,w]:=1;
>     elif T[i]=1 then
>       A[i,i]:=1;
>       A[i,w]:=-t;
>       A[i,G[i]]:=t-1;
>     else
>       A[i,i]:=-t;
>       A[i,w]:=1;
>       A[i,G[i]]:=t-1;
>     fi;
>   od;
#wypełnienie macierzy Alexandra
>   B:=minor(A,1,1);
#obliczanie pewnego minoru
>   f:=collect(det(B),t);
>   st:=ldegree(f);
>   wsp:=sign(coeff(f,t,st));
>   sort(collect(f*(wsp*t^(-st)),t),t);
#uzyskanie wielomianu w ostatecznej postaci
> end:
> wielomian(n,G,T);

```

n – liczba skrzyżowań,

G – wektor generatorów ($G[i]$ – numer generatora przechodzącego przez i -te skrzyżowanie),

T – wektor typów skrzyżowań ($T[i]$ – typ i -tego skrzyżowania).

5. Wyznaczenie wektorów G i T z dyskretnego zbioru punktów opisujących węzeł

W obliczeniach numerycznych węzeł dany jest w postaci dyskretnej jako zbiór n punktów (x_i, y_i, z_i) . Wyobraźmy sobie, że kolejne punkty połączone są odcinkami $[(x_i, y_i, z_i), (x_{i+1}, y_{i+1}, z_{i+1})]$, gdzie $i = 1, 2, \dots, n$, przy czym $(x_{n+1}, y_{n+1}, z_{n+1}) = (x_1, y_1, z_1)$. Aby obliczyć wielomian Alexandra dla węzła opisanego za pomocą zbioru punktów (x_i, y_i, z_i) , należy więc najpierw znaleźć wszystkie punkty skrzyżowań, ich typy i generatory przez nie przechodzące. Załóżmy, że mamy dany plik tekstowy, którego i -ty wiersz składa się z punktów x_i, y_i, z_i . Wtedy napisany w języku Maple program szukający skrzyżowań, ich typów i generatorów może mieć następującą postać:

```

> P:=readlib(readdata) (`knot.txt`, 3);
> n:=linalg[vectdim](P);
> wiad=[wszystko_dobrze, 0];

```

```

> for i to n do
>   if i=n then w:=1 else w:=i+1 fi;
>   p[i]:=0:
>   if i>2 then
>     for k to i-2 do
>       x1:=P[i][1];y1:=P[i][2];z1:=P[i][3];
>       x2:=P[w][1];y2:=P[w][2];z2:=P[w][3];
>       x3:=P[k][1];y3:=P[k][2];z3:=P[k][3];
>       x4:=P[k+1][1];y4:=P[k+1][2];z4:=P[k+1][3];
>       a:=[x2-x1,y2-y1,z2-z1];
>       b:=[x4-x3,y4-y3,z4-z3];
>       x[1]:=x1+t*a[1];
>       y[1]:=y1+t*a[2];
>       x[2]:=x3+s*b[1];
>       y[2]:=y3+s*b[2];
>       eq1:=solve({x[1]=x[2],y[1]=y[2]},{s,t});
>       t1:=subs(eq1,t);
>       s1:=subs(eq1,s);
>       d:=false;
>       if a[1]*b[2]<>b[1]*a[2] then
>         if (t1>0) and (t1<1) and (s1>0) and (s1<1) then
>           d:=true;
>         fi;
>         if d=true then
>           z[1]:=z1+t1*a[3];
>           z[2]:=z3+s1*b[3];
>           tp:=a[1]*b[2]-a[2]*b[1];
>           if z[2]>z[1] then
>             if p[i]<>0 then
>               wiad:=[blad,i];
>             fi;
>             p[i]:=1;
>             typ[i]:=sign(tp);
>             g[i]:=k;
>             if p[k]<>0 then
>               wiad:=[blad,k];
>             fi;
>             p[k]:=-1;
>           else
>             if p[k]<>0 then
>               wiad:=[blad,k];
>             fi;
>             p[k]:=1;
>             typ[k]:=-sign(tp);
>             g[k]:=i;
>             if p[i]<>0 then
>               wiad:=[blad,i];
>             fi;
>             p[i]:=-1;
>           fi;
>         fi;
>       fi;
>     od;
>   fi;
> od:


---


> wiad;
> j:=0;
> for i to n do
>   if p[i]=1 then
>     j:=j+1;
>     punkt_prze[j]:=i;

```

```

>     pom[j]:=g[i];
>     if typ[i]=1 then
>         T[j]:=1;
>     else
>         T[j]:=2;
>     fi;
> fi;
> od;
> for i to j do
>     if i=1 then w:=j else w:=i-1 fi;
>     gen[i]:=[punkt_prze[w],punkt_prze[i]];
> od;
> for i to j do
>     for k to j do
>         if k<>1 then
>             if (gen[k][1]<pom[i]) and (pom[i]<gen[k][2]) then
>                 G[i]:=k;
>             fi;
>             elif (pom[i]>gen[k][1]) or (pom[i]<gen[k][2]) then
>                 G[i]:=k;
>             fi;
>         od;
>     print(i,G[i],T[i]);
> od;

```

Powyższy program realizuje następujący algorytm:

1. Przyjmujemy kierunek wzdłuż węzła zgodnie z uporządkowaniem punktów.
2. Dla każdego punktu $P[i] = (x_i, y_i, z_i)$, $i = 3, 4, \dots, n$, piszemy równanie prostej $l[i]$ w postaci parametrycznej łączącej go z punktem $P[i+1]$ i sprawdzamy, czy przecina się z prostą $l[k]$, łączącą punkty $P[k]$ i $P[k+1]$ dla $k = 1, 2, \dots, i-2$ w punkcie leżącym między punktami $P[i]$ i $P[i+1]$ oraz $P[k]$ i $P[k+1]$.

Prosta $l[i]$ w postaci parametrycznej ma postać

$$\begin{aligned} x &= x_i + t a_1, \\ y &= y_i + t a_2, \\ z &= z_i + t a_3, \end{aligned}$$

gdzie wektor kierunkowy prostej $\mathbf{a} = [a_1, a_2, a_3] = [x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i]$.

Podobnie prostą $l[k]$ zapiszemy w postaci

$$\begin{aligned} x &= x_k + s b_1, \\ y &= y_k + s b_2, \\ z &= z_k + s b_3, \end{aligned}$$

gdzie wektor kierunkowy prostej $\mathbf{b} = [b_1, b_2, b_3] = [x_{k+1} - x_k, y_{k+1} - y_k, z_{k+1} - z_k]$.

Przecięcie krzywej będącej projekcją nastąpi, gdy układ

$$\begin{cases} x_i + t a_1 = x_k + s b_1, \\ y_i + t a_2 = y_k + s b_2 \end{cases}$$

ma rozwiązania spełniające warunek $0 < s_{ik}, t_{ik} < 1$. Zakładamy tutaj, że żaden punkt $P[i]$ nie jest skrzyżowaniem.

3. Załóżmy, że p. 2 jest spełniony. Wtedy zmiennej $p[i]$ przypisujemy wartość 1, jeżeli

prosta $l[i]$ leży pod prostą $l[k]$ (tzn. $z_i + t_{ik} a_3 < z_k + s_{ik} b_3$). W przeciwnym przypadku przyjmujemy $p[i] = -1$. Oczywiście, jeśli $p[i] = 1$, to $p[k] = -1$, natomiast jeśli $p[i] = -1$, to $p[k] = 1$. Wprowadźmy nową zmienną g taką, że jeśli $p[i] = 1$, to $g[i] = k$ lub jeśli $p[k] = 1$, to $g[k] = i$.

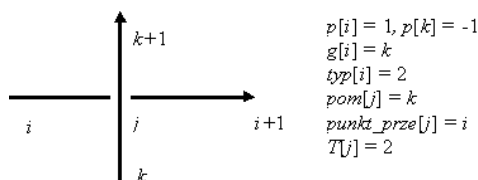
- Określamy znak iloczynu wektorowego wektorów \mathbf{a} i \mathbf{b} . Jeśli $p[i] = 1$, $typ[i] = \text{sign}(\mathbf{a} \times \mathbf{b})$, w przeciwnym razie $typ[k] = \text{sign}(\mathbf{b} \times \mathbf{a})$. Zmienna typ pozwoli nam określić typ skrzyżowania.

Punkty 1 – 4 pozwalają określić wszystkie skrzyżowania się krzywej będącej rzutem wężła na płaszczyznę xy .

- Następnie zmieniając i od 1 do n nadajemy punktom, dla których $p[i] = 1$, numery od 1 do s , gdzie s oznacza liczbę skrzyżowań (odpowiada to numerowaniu skrzyżowań w algorytmie Alexandra). Pod zmienną $punkt_prze[j]$ ($j = 1, 2, \dots, s$) podstawiamy numer punktu i , natomiast pod zmienną $pom[j]$ podstawiamy $g[i]$. Zmienna $T[j]$ określa typ skrzyżowania: $T[j] = 1$, gdy $typ[i] = 1$ lub $T[j] = 2$, jeśli $typ[i] = -1$.
- Dzielimy krzywą na generatory. Zgodnie z wymaganiami algorytmu Alexandra przez $gen[j]$ oznaczmy przedział $[punkt_prze[j-1], punkt_prze[j]]$.
- Pozostało określić $G[j]$ – numer generatora przechodzącego przez j -te skrzyżowanie. W tym celu sprawdzamy, do którego generatora należy wartość $pom[j]$.

W algorytmie wartość $n + 1 \rightarrow 1$. Nie należy mylić numeru punktu z numerem skrzyżowania.

W ten sposób wyznaczyliśmy liczbę skrzyżowań, wektor ich typów ($T[j]$) oraz generatorów przez nie przechodzących ($G[j]$).



Rys. 3. Wartości przykładowych zmiennych w przypadku skrzyżowania się krzywej $l[i]$ z $l[k]$ (j oznacza numer skrzyżowania)

Jeżeli istnieje taki punkt o numerze i , że między $P[i]$ oraz $P[i + 1]$ nastąpi dwukrotne (lub kilkakrotne) skrzyżowanie krzywej będącej rzutem wężła na płaszczyznę xy , powyższy algorytm nie zadziała poprawnie. Zmienna $wiad$ zasygnalizuje, że obliczenia nie będą poprawne, a ponadto poda numer punktu i , który nie spełnia wymagań algorytmu. W takim przypadku konieczne jest dodanie punktów opisujących wężel.

6. Tablica wielomianów Alexandra węzłów o $MCN = 9$

W tabeli 1 podano tablicę wielomianów Alexandra dla węzłów o minimalnej liczbie skrzyżowań równej 9. Wielomiany te obliczono za pomocą programów podanych w p. 4 i 5.

Tabela 1. Tabela wielomianów Alexandra dla węzłów o liczbie skrzyżowań 9

Typ	Wielomian	Typ	Wielomian
9_1	$t^8 - t^7 + t^6 - t^5 + t^4 - t^3 + t^2 - t + 1$	9_{26}	$t^6 - 5t^5 + 11t^4 - 13t^3 + 11t^2 - 5t + 1$
9_2	$4t^2 - 7t + 4$	9_{27}	$t^6 - 5t^5 + 11t^4 - 15t^3 + 11t^2 - 5t + 1$
9_3	$2t^6 - 3t^5 + 3t^4 - 3t^3 + 3t^2 - 3t + 2$	9_{28}	$t^6 - 5t^5 + 12t^4 - 15t^3 + 12t^2 - 5t + 1$
9_4	$3t^4 - 5t^3 + 5t^2 - 5t + 3$	9_{29}	$t^6 - 5t^5 + 12t^4 - 15t^3 + 12t^2 - 5t + 1$
9_5	$6t^2 - 11t + 6$	9_{30}	$t^6 - 5t^5 + 12t^4 - 17t^3 + 12t^2 - 5t + 1$
9_6	$2t^6 - 4t^5 + 5t^4 - 5t^3 + 5t^2 - 4t + 2$	9_{31}	$t^6 - 5t^5 + 13t^4 - 17t^3 + 13t^2 - 5t + 1$
9_7	$3t^4 - 7t^3 + 9t^2 - 7t + 3$	9_{32}	$t^6 - 6t^5 + 14t^4 - 17t^3 + 14t^2 - 6t + 1$
9_8	$2t^4 - 8t^3 + 11t^2 - 8t + 2$	9_{33}	$t^6 - 6t^5 + 14t^4 - 19t^3 + 14t^2 - 6t + 1$
9_9	$2t^6 - 4t^5 + 6t^4 - 7t^3 + 6t^2 - 4t + 2$	9_{34}	$t^6 - 6t^5 + 16t^4 - 23t^3 + 16t^2 - 6t + 1$
9_{10}	$4t^4 - 8t^3 + 9t^2 - 8t + 4$	9_{35}	$7t^2 - 13t + 7$
9_{11}	$t^6 - 5t^5 + 7t^4 - 7t^3 + 7t^2 - 5t + 1$	9_{36}	$t^6 - 5t^5 + 8t^4 - 9t^3 + 8t^2 - 5t + 1$
9_{12}	$2t^4 - 9t^3 + 13t^2 - 9t + 2$	9_{37}	$2t^4 - 11t^3 + 19t^2 - 11t + 2$
9_{13}	$4t^4 - 9t^3 + 11t^2 - 9t + 4$	9_{38}	$5t^4 - 14t^3 + 19t^2 - 14t + 5$
9_{14}	$2t^4 - 9t^3 + 15t^2 - 9t + 2$	9_{39}	$3t^4 - 14t^3 + 21t^2 - 14t + 3$
9_{15}	$2t^4 - 10t^3 + 15t^2 - 10t + 2$	9_{40}	$t^6 - 7t^5 + 18t^4 - 23t^3 + 18t^2 - 7t + 1$
9_{16}	$2t^6 - 5t^5 + 8t^4 - 9t^3 + 8t^2 - 5t + 2$	9_{41}	$3t^4 - 12t^3 + 19t^2 - 12t + 3$
9_{17}	$t^6 - 5t^5 + 9t^4 - 9t^3 + 9t^2 - 5t + 1$	9_{42}	$t^4 - 2t^3 + t^2 - 2t + 1$
9_{18}	$4t^4 - 10t^3 + 13t^2 - 10t + 4$	9_{43}	$t^6 - 3t^5 + 2t^4 - t^3 + 2t^2 - 3t + 1$
9_{19}	$2t^4 - 10t^3 + 17t^2 - 10t + 2$	9_{44}	$t^4 - 4t^3 + 7t^2 - 4t + 1$
9_{20}	$t^6 - 5t^5 + 9t^4 - 11t^3 + 9t^2 - 5t + 1$	9_{45}	$t^4 - 6t^3 + 9t^2 - 6t + 1$
9_{21}	$2t^4 - 11t^3 + 17t^2 - 11t + 2$	9_{46}	$2t^2 - 5t + 2$
9_{22}	$t^6 - 5t^5 + 10t^4 - 11t^3 + 10t^2 - 5t + 1$	9_{47}	$t^6 - 4t^5 + 6t^4 - 5t^3 + 6t^2 - 4t + 1$
9_{23}	$4t^4 - 11t^3 + 15t^2 - 11t + 4$	9_{48}	$t^4 - 7t^3 + 11t^2 - 7t + 1$
9_{24}	$t^6 - 5t^5 + 10t^4 - 13t^3 + 10t^2 - 5t + 1$	9_{49}	$3t^4 - 6t^3 + 7t^2 - 6t + 3$
9_{25}	$3t^4 - 12t^3 + 17t^2 - 12t + 3$		

Z powyższej tabeli wynika, że węzły 9_{29} i 9_{28} mają identyczne wielomiany. Takie same wielomiany istnieją również dla następujących par węzłów: $(6_1, 9_{46})$, $(7_4, 9_2)$, $(8_{14}, 9_8)$, $(8_{18}, 9_{24})$.

Podziękowania

Dziękujemy Panu Andrzejowi Stasiakowi za udostępnienie nam literatury dotyczącej algorytmu wyznaczania wielomianu Alexandra.

Literatura

- [1] Katritch, V. (et al), *Nature* 384 (1996), 142.
- [2] Pierański, P., *Pro Dialog* 5 (1996), 111.
- [3] Adams, C. C., *The Knot Book*, Freemann and Co., New York 1994.
- [4] Kamenetskij, M. D., Vologodskij, A. V., *Sov. Phys. Usp.* 24 (1981), 679.