

O gładkim łączeniu krzywych Béziera

1. Wstęp

Podstawowym budulcem systemów graficznych operujących na płaszczyźnie (np. edytorów fontowych) są krzywe sklejane, zwane też krzywymi giętymi lub – z angielska – splajnami. Na ogół są to wielomiany jednej zmiennej, powiedzmy t . Zwyczajowo przyjmuje się, że $0 \leq t \leq 1$. Oznacza to, że zwykle operuje się na fragmentach krzywych (łukach). Skomplikowane kształty, jakie są niezbędne w praktycznych zastosowaniach, uzyskuje się, łącząc odpowiednio fragmenty krzywych.

Niniejsze opracowanie jest poświęcone dwóm technikom łączenia krzywych opisywanych wielomianami 3 stopnia, zwanych krzywymi Béziera i adresowane jest do Czytelników obeznanych praktycznie z krzywymi Béziera, acz niekoniecznie zajmujących się na co dzień matematyką. Wprawdzie niezbędna jest znajomość matematyki na poziomie elementarnym, jednakże dla jak największej jasności wyводу nawet proste fakty matematyczne są uzasadniane i – gdzie to tylko możliwe – podawana jest interpretacja geometryczna tych faktów.

2. Krzywe sklejane

W praktyce spotyka się najczęściej dwa rodzaje krzywych sklejanych: krzywe zwane *krzywymi B-sklejanymi* (zob. [9]) oraz *krzywe Béziera* (zob. [10]); oba rodzaje są w istocie szczególnym przypadkiem wielomianów Bernsteina na płaszczyźnie (zob. [11]), odpowiednio 2 i 3 stopnia. Wielomiany Bernsteina stopnia n zdefiniowane są za pomocą następującego wzoru:

$$P(t) \stackrel{\text{def}}{=} \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i, \quad (1)$$

gdzie P_i są punktami na płaszczyźnie. W przypadku krzywych B-sklejanych i krzywych Béziera powyższa formuła przybiera odpowiednio postać:

$$A(t) = (1-t)^2 A_0 + 2t(1-t) A_1 + t^2 A_2, \quad (2)$$

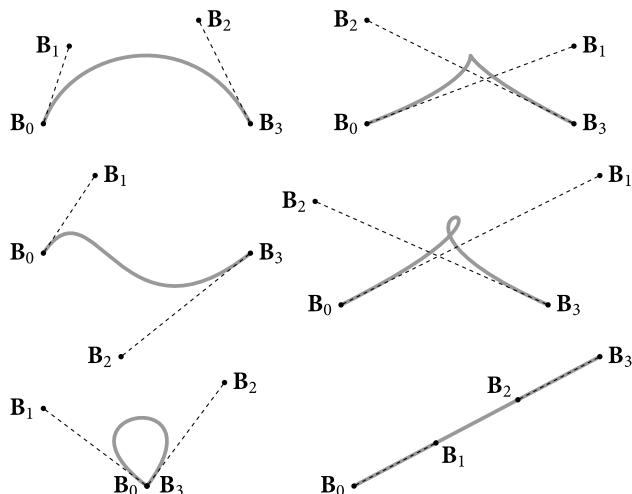
$$B(t) = (1-t)^3 B_0 + 3t(1-t)^2 B_1 + 3t^2(1-t) B_2 + t^3 B_3. \quad (3)$$

Zauważmy od razu, że krzywa B-sklejana daje się w trywialny sposób zapisać jako krzywa Béziera. Łatwo sprawdzić, że należy w tym celu przyjąć następujące zależności (traktując punkty na płaszczyźnie jako wektory wodzące):

$$B_0 \stackrel{\text{def}}{=} A_0, \quad B_1 \stackrel{\text{def}}{=} A_0 + \frac{2}{3}(A_1 - A_0), \quad B_2 \stackrel{\text{def}}{=} A_2 + \frac{2}{3}(A_1 - A_2), \quad B_3 \stackrel{\text{def}}{=} A_2. \quad (4)$$

Wzory (4) przydadzą się nam punkcie 4.

Oczywiście krzywe B-sklejane i krzywe Béziera nie są jedynymi krzywymi używanymi w programach graficznych. Godne odnotowania są np. *krzywe spiro* (zob. [5]) będące fragmentami klotoidy (zwanej również spiralą Eulera lub spiralą Cornu).

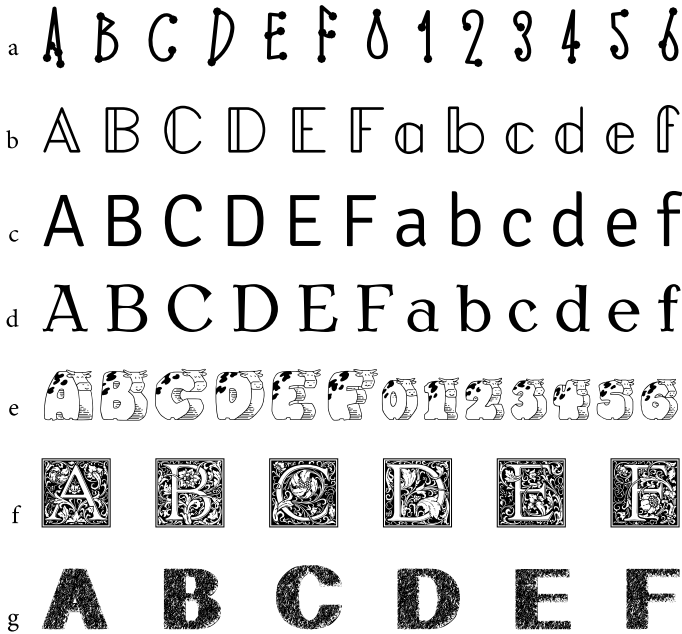


Rys. 1. Krzywe Béziera uzyskane za pomocą wzoru (3) ($0 \leq t \leq 1$); pominięty został przypadek zdegenerowany, $B_0 = B_1 = B_2 = B_3$, opisujący pojedynczy punkt

W dalszym ciągu niniejszego opracowania skupimy się na krzywych Béziera, obsługiwanych przez praktycznie wszystkie programy graficzne (zarówno komercyjne, jak i openwarowe). Krzywą opisywaną pojedynczym wielomianem Béziera będziemy określać jako segment lub – zamiennie – łuk Béziera. Punkty B_0 i B_3 będziemy nazywać węzłami, a punkty B_1 i B_2 – naciągami (konkretne oznaczenia literowe nie są istotne).

Za pomocą wielomianu Béziera, określonego równaniem (3), można opisać względnie niewiele istotnie różnych kształtów. Rysunek 1 przedstawia sześć wizualnie różnych form, jakie przybiera łuk Béziera.

Jest oczywiste, że za pomocą pojedynczego łuku Béziera nawet tak proste kształty, jak wielokąty nie dają się opisać. Jednakże łączenie wielu segmentów w jedną krzywą stwarza ogromne możliwości, co ilustruje rysunek 2, prezentujący gamę przykładów – od fontów o kształtach zbudowanych z niemalże prymitywnych figur geometrycznych poprzez typowe pisma dzielowe (szeryfowe i bezszeryfowe) do fontów złożonych z bardzo skomplikowanych graficznie znaków. Przykłady (e), (f) i (g) należałoby zaliczyć do obszaru grafiki komputerowej; użycie do tego rodzaju znaków technologii fontowej jest w zasadzie nadużyciem, zwłaszcza w przypadku fontu *Geometric Petras*, który sprawia kłopoty niemal wszystkim programom przetwarzającym ten font.



Rys. 2. Przykłady znaków, opisanych za pomocą wielomianów Bernsteina stopnia 2 i 3, pochodzących z następujących fontów: (a) *Punk Nova* Donalda E. Knutha (rozszerzenie zestawu znaków i konwersja do formatu OTF – Taco Hoekwater, Hans Hagen i Khaled Hosny); (b) *Blackboard Bold* Alana Jeffreya; (c) *Signika* Anny Giedryś; (d) *Antykwa Toruńska* Zygryda Gardzielewskiego (dygitalizacja – Janusz M. Nowacki); (e) *koeieletters* Duane’a Bibby’ego (dygitalizacja – Taco Hoekwater i Hans Hagen); (f) *Goudy Initialen* Frederika Goudy’ego (dygitalizacja – Dieter Steffmann); (g) *Geometric Petras* sygnowany Paulo W/ Intellecta Design

Znaki w fontach zdefiniowane są jako zbiory krzywych zamkniętych – co jest wymogiem obecnej komputerowej technologii fontowej (zob. [7, 8]) – z których każda zbudowana jest z połączonych w jedną krzywą wielomianów Bernsteina (stopnia 2 lub 3). Jak więc widać, za pomocą krzywych sklejanых można opisać obiekty graficznie o bardzo skomplikowanych kształtach.

Przytoczone przykłady dotyczyły fontów, należy jednak podkreślić, że krzywe sklepane z powodzeniem wykorzystuje się także w innych obszarach grafiki komputerowej, nawet trójwymiarowej, po uogólnieniu pojęcia jednowymiarowej krzywej sklepanej na płaty dwuwymiarowe.

Jak łatwo się domyślić, kluczowym zabiegiem przy konstruowaniu krzywych sklejanых jest właśnie „klejenie”, czyli łączenie poszczególnych fragmentów krzywej sklepanej według wybranych zasad. Najlepiej jest, gdy takie zasady mają precyzyjny opis matematyczny. W numerze 1 „Acta Poligraphica” (zob. [4]) opisana została szczegółowo metoda Johna D. Hobby’ego łączenia krzywej sklepanej, przechodzącej przez zadane punkty, uwzględniająca wszystkie zadane punkty (w tym sensie jest to metoda globalna). Omówione i porównane z metodą Hobby’ego zostały metody lokalne, uwzględniające bezpośrednie otoczenie punktu łączenia. Jak

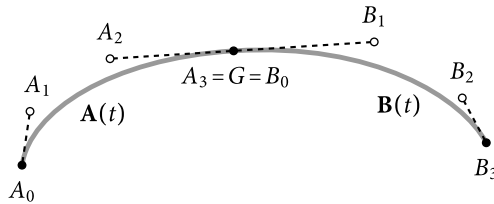
się okazuje, lokalne metody w niektórych zastosowaniach prowadzą do porównywalnych lub wręcz lepszych wyników niż globalna metoda Hobby'ego, a są przy tym prostsze w implementacji.

Reszta niniejszego artykułu poświęcona jest dwóm lokalnym technikom tworzenia gładkich połączeń krzywych Béziera, nadającym się do zaimplementowania interakcyjnego.

3. Znajdowanie gładkiego połączenia

Załóżmy, że mamy dane dwa łuki Béziera: $\mathbf{A}(t) = (1-t)^3 A_0 + 3t(1-t)^2 A_1 + 3t^2(1-t) A_2 + t^3 A_3$ oraz $\mathbf{B}(t) = (1-t)^3 B_0 + 3t(1-t)^2 B_1 + 3t^2(1-t) B_2 + t^3 B_3$. Oczywiście $\mathbf{A}(0) = A_0$, $\mathbf{A}(1) = A_3$, $\mathbf{B}(0) = B_0$, $\mathbf{B}(1) = B_3$. Dla uproszczenia rozważań załóżmy dodatkowo, że $A_i \neq B_j$ dla $i, j = 0, 1, 2, 3$, z wyjątkiem $A_3 = B_0$; to znaczy zakładamy, że rozważane dwa łuki tworzą ciągłą, dwusegmentową krzywą Béziera.

Będziemy starali się znaleźć takie położenie punktu $A_3 = G = B_0$, by połączenie było jak najgładsze. Mówiąc precyzyjnie, będziemy żądać, by lewo- i prawostronna krzywizna w punkcie G była taka sama. Założenia zadania zostały zilustrowane na rysunku 3.



Rys. 3. Sformułowanie problemu: przy zadanych punktach A_0, A_1, A_2, B_1, B_2 i B_3 należy wyznaczyć punkt G taki, że łuki $\mathbf{A}(t)$ i $\mathbf{B}(t)$ (przy czym $A_3 = G = B_0$) łączą się w tym punkcie możliwie gładko

Zauważmy, że założenie gładkości połączenia w punkcie G implikuje współliniowość punktów A_2, G i B_1 . Wynika to z podstawowej własności krzywych Béziera, mianowicie linie proste wyznaczone przez pary punktów $A_0 A_1, A_2 A_3, B_0 B_1$ oraz $B_2 B_3$ są styczne do odpowiednich łuków Béziera w punktach A_0, A_3, B_0 i B_3 . Przyjmijmy zatem, że warunkiem, który musi spełniać punkt G , jest leżenie na linii prostej wyznaczonej przez punkty A_2 i B_1 .

Dodatkowo będziemy żądać, żeby krzywizna łuku $\mathbf{A}(t)$ dla $t = 1$ była równa krzywiznie łuku $\mathbf{B}(t)$ dla $t = 0$. *Krzywizną* w danym definiuje się jako odwrotność promienia okręgu stycznego do krzywej w danym punkcie; promień tego okręgu nosi nazwę *promienia krzywizny*.

Zależności pozwalające obliczyć krzywiznę wielomianu Bernsteina w danym punkcie można znaleźć m.in. we wspomnianej pracy [4]. Tu przytoczymy odnośne wzory bez dowodu.

Niech $|v|$ dla danego wektora (2-wymiarowego) $v = (v_x, v_y)$ oznacza jego długość, tzn. $|v| \stackrel{\text{def}}{=} \sqrt{v_x^2 + v_y^2}$. Ponadto zdefiniujmy dla pary wektorów $u = (u_x, u_y)$ i $v = (v_x, v_y)$ ich wyznacznik (ang. *determinant*) jako $\det(u, v) \stackrel{\text{def}}{=} u_x v_y - u_y v_x$. Je-

zeli potraktujemy punkty na płaszczyźnie jako wektory wodzące (tzn. potraktujemy współrzędne punktów jako składowe wektora), możemy wzory na krzywiznę κ_A i κ_B dla łuków Béziera odpowiednio $\mathbf{A}(t)$ i $\mathbf{B}(t)$ w punkcie G (czyli $\mathbf{A}(1) = \mathbf{B}(0)$) zapisać następująco:

$$\kappa_A \stackrel{\text{def}}{=} \frac{2 \det(G - A_2, A_1 - A_2)}{3 |G - A_2|^3}, \quad \kappa_B \stackrel{\text{def}}{=} \frac{2 \det(B_1 - G, B_2 - B_1)}{3 |G - B_1|^3}. \quad (5)$$

Zauważmy, że punkty A_0 i B_3 nie występują w powyższych formułach.

Zdefiniujmy wielkości h_A i h_B następująco (ich sens geometryczny zostanie wyjaśniony dalej):

$$h_A \stackrel{\text{def}}{=} \frac{\det(G - A_2, A_1 - A_2)}{|G - A_2|}, \quad h_B \stackrel{\text{def}}{=} \frac{\det(B_1 - G, B_2 - B_1)}{|G - B_1|}. \quad (6)$$

Dalej pokażemy, że wielkości h_A i h_B nie zależą od położenia punktu G na prostej przechodzącej przez A_2 i B_1 i że do ich wyznaczenia wystarczy znajomość wielkości A_1, A_2, B_1 i B_2 .

Oznaczmy nieznanne wielkości $|G - A_2|$ i $|G - B_1|$ jako odpowiednio p i q . Zauważmy, że $p + q = |A_2 - B_1|$. Warunek równości krzywizn $\kappa_A = \kappa_B$ można zatem przeformułować następująco:

$$\frac{h_A}{p^2} = \frac{h_B}{q^2}, \quad (7)$$

a tym samym zadanie redukuje się do układu dwóch równań liniowych z dwoma niewiadomymi p i q :

$$\frac{p}{q} = \sqrt{\frac{h_A}{h_B}}, \quad p + q = |A_2 - B_1|. \quad (8)$$

Powyższe wzory nie dają się zastosować w przypadku $h_A h_B \leq 0$. Nie będziemy tego zagadnienia rozwijać, poprzestaniemy na stwierdzeniu, że odpowiada on przyjęciu, że punkt G jest punktem przegięcia i uwzględnienie tego przypadku jest możliwe i nietrudne.

Pozostaje zatem wyznaczyć wielkości h_A i h_B .

Korzystając z definicji wyznacznika, łatwo można sprawdzić, że zależności (6) są równoważne poniższym zależnościom:

$$h_A = \det\left(\frac{G - A_2}{|G - A_2|}, A_1 - A_2\right), \quad h_B = \det\left(\frac{B_1 - G}{|G - B_1|}, B_2 - B_1\right). \quad (9)$$

Oczywiście wektory $(G - A_2)/|G - A_2|$ i $(B_1 - G)/|B_1 - G|$ pojawiające się w równaniach (9) mają jednostkową długość. Ponieważ zgodnie z założeniem punkty A_2, G i B_1 są współliniowe, zachodzą następujące zależności:

$$\frac{G - A_2}{|G - A_2|} = \frac{B_1 - G}{|G - B_1|} = \frac{B_1 - A_2}{|A_2 - B_1|}. \quad (10)$$

Tym samym zależności (6) można zapisać, jak następuje:

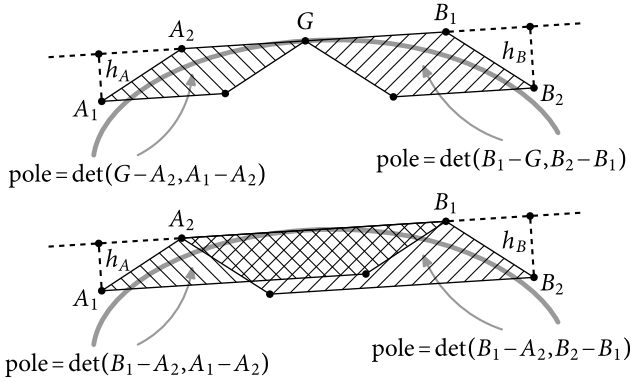
$$h_A = \frac{\det(B_1 - A_1, A_1 - A_2)}{|B_1 - A_2|}, \quad h_B = \frac{\det(B_1 - A_1, B_2 - B_1)}{|B_1 - A_2|}, \quad (11)$$

co łatwo sprawdzić bezpośrednim rachunkiem. I to w istocie kończy procedurę wyznaczania punktu G – wzory (8) i (11) pozwalają w prosty sposób obliczyć współrzędne punktu G .

Zauważmy, że we wzorach (8) może się pojawić nieokreśloność, gdy punkty A_1, A_2, B_1 i B_2 są współliniowe, bowiem wówczas h_A i h_B przyjmują równocześnie wartość zero i w konsekwencji położenie punktu G nie jest jednoznacznie określone. Jeśli tylko trzy z tych punktów są współliniowe, mamy do czynienia z rozwiązaniem zdegenerowanym (punkt G pokrywa się z punktem A_2 bądź B_1).

3.1. Interpretacja geometryczna

Geometryczna interpretacja przekształcenia formuł (6) do postaci (11) wynika z geometrycznych własności wyznacznika. Mianowicie wartość wyznacznika $\det(u, v)$ jest równa wielkości pola równoległoboku wyznaczonego przez wektory u i v (tak rozumiane pole ma znak zależny od wzajemnej orientacji wektorów). Zatem wzory (6) definiują wysokości odpowiednich równoległoboków i – jak wynika z rysunku 4 – są to te same wysokości, tyle że równoległoboków określonych przez wektory występujące we wzorach (11).

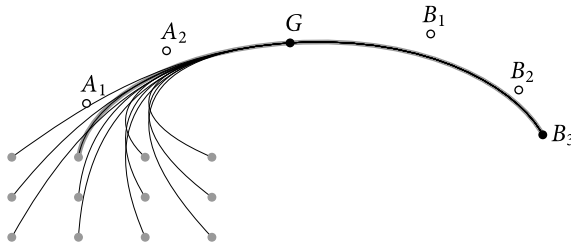


Rys. 4. Geometryczne uzasadnienie równoważności wzorów (6) i (11)

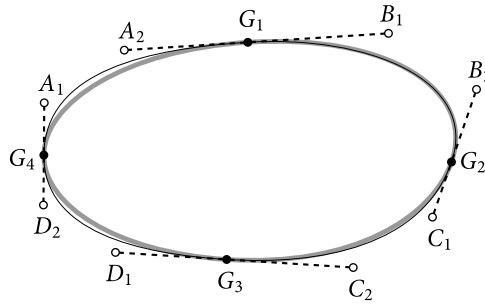
Reasumując, rysunek 4 stanowi geometryczny dowód na to, że wielkości h_A i h_B określone wzorami (6) i (11) są identyczne.

3.2. Przykłady

Jak zostało wspomniane, skrajne węzły krzywych Béziera nie pojawiają się ani we wzorach (8), ani we wzorach (11), co oznacza, że położenie punktu G nie



Rys. 5. Położenie punktu gładkiego połączenia G nie zmienia się na skutek zmiany położenia punktu A_0 (szare kółka), podobnie zresztą, jak i zmiany położenia punktu B_3



Rys. 6. Zadanie parzystej liczby naciągów (tu: $A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2$) wyznacza jednoznacznie krzywą zamkniętą – linia szara; czarną linią zaznaczona została krzywa przechodząca przez punkty G_1, G_2, G_3 i G_4 , obliczona za pomocą interpolacyjnego algorytmu Hobby'ego (dz. cyt.)

zależy od położenia skrajnych węzłów rozważanych łuków Béziera. Rysunek 5 ilustruje ten fakt.

Innymi słowy, położenie punktów skrajnych krzywej otwartej można ustalić dowolnie (np. na podstawie dodatkowych warunków). W przypadku krzywej wielosegmentowej wewnętrzne segmenty są określone jednoznacznie, skąd wynika, że w przypadku krzywej zamkniętej podanie położenia naciągów jednoznacznie opisuje całą krzywą, jak to przedstawia rysunek 6.

3.3. Podsumowanie

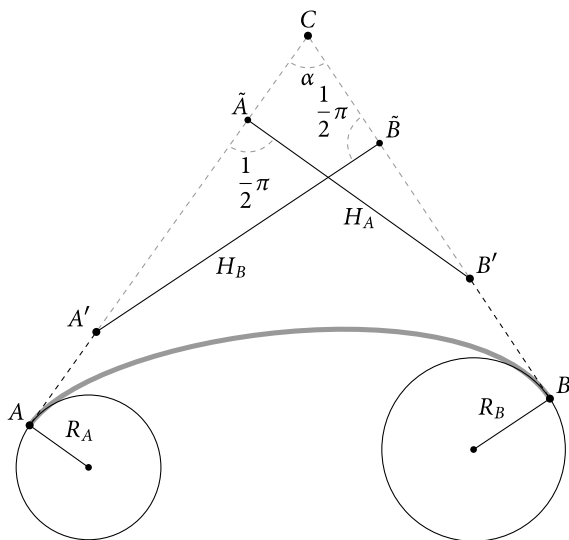
Powyższe rozważania wyjaśniają, w jaki sposób mając zadane naciągi łuków Béziera, można te łuki połączyć z zachowaniem stałej krzywizny w punktach łączenia. Jest to w pewnym sensie podejście dualne do podejścia Hobby'ego: w metodzie Hobby'ego zadaje się węzły (plus ewentualnie dodatkowe warunki) i wyznacza naciągi. W opisaney wyżej metodzie zadaje się naciągi (bez dodatkowych warunków) i wyznacza węzły.

Metoda ta doczekała się obiecującej, chociaż ciągle jeszcze eksperymentalnej, implementacji w komercyjnym edytorze fontów FONTLAB.

4. Łuk Béziera o zadanej krzywiznie w punktach skrajnych

Jak poprzednio, założymy (zmieniając nieco oznaczenia na nieco wygodniejsze dla tego zadania), że dana jest krzywa Béziera: $\mathbf{P}(t) = (1-t)^3A + 3t(1-t)^2A' + 3t^2(1-t)B' + t^3B$. Dla uproszczenia przyjmiemy, że punkty A, A', B' i B są parami różne. Dodatkowo założymy, że linie proste wyznaczone przez pary punktów AA' i BB' przecinają się pod kątem α , przy czym $\alpha < \pi$. Na rysunku 7 przedstawiona została założona sytuacja oraz wprowadzone zostały punkty pomocnicze \bar{A}, \bar{B} i wielkości dodatkowe H_A, H_B, R_A, R_B .

Odległości H_A i H_B będziemy uważać za wielkości ze znakiem, przyjmując następującą konwencję: jeśli punkt B' leży na lewo od linii prostej skierowanej od A do A' , to $H_B < 0$, w przeciwnym razie $H_B > 0$; analogicznie, jeśli punkt A' położony na prawo od linii prostej skierowanej od B' do B , to $H_A < 0$, w przeciwnym



Rys. 7. Graficzne przedstawienie przyjętych założeń

razie $H_A > 0$. Dla przypadku przedstawionego na rysunku 7 obie wartości są ujemne: $H_A < 0$, $H_B < 0$. Również kąty, w szczególności kąt α , będziemy uważać za wielkości ze znakiem, mianowicie kąty skierowane zgodnie z kierunkiem ruchu wskazówek zegara uznamy za ujemne; w naszym przypadku kąt α skierowany przeciwnie do ruchu wskazówek zegara od CA do CB , zatem $\alpha > 0$. (Zauważmy, że przyjęta konwencja pozostaje w zgodzie z podaną w punkcie 3 definicją wyznacznika i w istocie jest konsekwencją tej definicji.)

Okręgi obrazują krzywiznę w punktach A i B ; R_A i R_B oznaczają odpowiednie promienie krzywizny.

4.1. Sformułowanie problemu

Zacniemy od precyzyjnego matematycznie sformułowania zadania:

PROBLEM PROMIENI KRZYWIZNY: Niech dany będzie trójkąt ABC oraz liczby rzeczywiste R_A i R_B ; należy wyznaczyć położenie naciągów A' i B' należących odpowiednio do boków AC i BC trójkąta ABC tak, by promień krzywizny łuku Béziera (opartego na punktach $AA'B'B$) wynosił R_A i R_B w punktach odpowiednio A i B .

Pamiętając, że promień krzywizny definiowany jest jako odwrotność krzywizny, wzory (5) możemy zapisać następująco:

$$R_A = \frac{3}{2} \frac{|AA'|^2}{H_A}, \quad R_B = \frac{3}{2} \frac{|BB'|^2}{H_B}. \quad (12)$$

Wprowadźmy oznaczenia $l_A = |AA'|$, $L_A = |AC|$, $l_B = |BB'|$, $L_B = |BC|$; wówczas (uwzględniając znaki H_A , H_B i α) $H_A = -(L_B - l_B) \sin \alpha$, $H_B = -(L_A - l_A) \sin \alpha$. Tym

samym równania (12) można przepisać w postaci:

$$R_A = -\frac{3}{2} \frac{l_A^2}{(L_B - l_B) \sin \alpha}, \quad R_B = -\frac{3}{2} \frac{l_B^2}{(L_A - l_A) \sin \alpha}, \quad (13)$$

lub równoważnie:

$$l_B = L_B + \frac{3}{2} \frac{l_A^2}{R_A \sin \alpha}, \quad l_A = L_A + \frac{3}{2} \frac{l_B^2}{R_B \sin \alpha}. \quad (14)$$

Zależności (14) posłużą nam do rozwiązania postawionego zadania.

4.2. Parabola jako krzywa Béziera

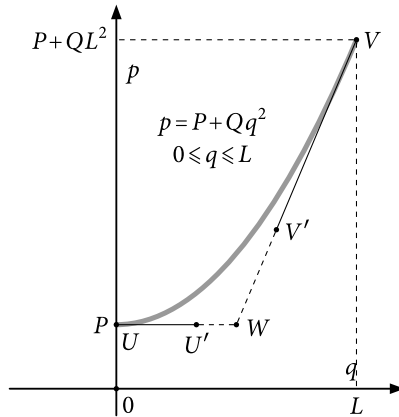
Równania (14) można zinterpretować jako definicje dwóch parabol o osiach wzajemnie prostopadłych (zmiennych l_A i l_B). Ponieważ założyliśmy, że punkty A' i B' leżą na bokach trójkąta ABC , dokładniej $0 \leq l_A \leq L_A$ oraz $0 \leq l_B \leq L_B$, wzory te w istocie opisują półparabole; punkt przecięcia półparabol określa rozwiązanie sformułowanego wyżej *problemu promieni krzywizny* (por. punkt 4.3).

Jak zostało odnotowane w punkcie 2, parabole w trywialny sposób dają się przedstawić jako krzywe Béziera. Dostosujmy wzory (4) do naszego przypadku. Przypuśćmy, że na płaszczyźnie qp mamy zadane równanie (fragmentu) półparaboli $q = Q + Pp^2$, $0 \leq p \leq L$. Odpowiednie węzły krzywej Béziera obliczyć można ze wzorów

$$\text{węzły: } U = (0, P), \quad V = (L, P + QL^2) \quad (15)$$

$$\text{naciągi: } U' = \left(\frac{1}{3}L, P\right), \quad V' = \left(\frac{2}{3}L, P + \frac{1}{3}QL^2\right) \quad (16)$$

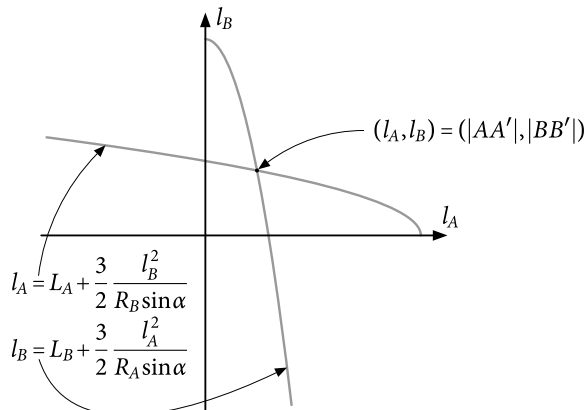
Geometryczną interpretację powyższych formuł przedstawia rysunek 8. Przedstawiona na rysunku krzywa może być zapisana jako $(t - 1)^2U + 2t(t - 1)W + t^2V$, gdzie $W = \left(\frac{1}{2}L, P\right)$, bądź $(t - 1)^3U + 3t(t - 1)^2U' + 3t^2(t - 1)V' + t^3V$.



Rys. 8. Przedstawienie fragmentu półparaboli jako krzywej Béziera

4.3. Rozwiązanie numeryczne – zarys

Zatem zadanie zostało sprowadzone do znajdowania punktu (ogólnie – punktów) przecięcia dwóch półparabol. Rysunek 9 przedstawia odpowiednie półparabole obliczonego według wzorów (14) dla przypadku przedstawionego na rysunku 7 (przypomnijmy, że w tym przypadku $R_A < 0$, $R_B < 0$, $\alpha > 0$).



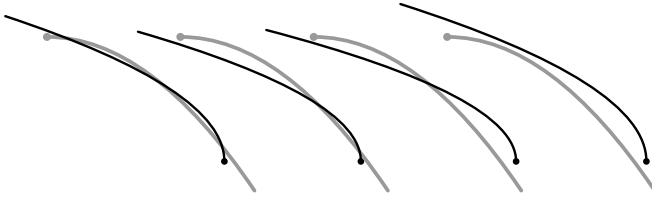
Rys. 9. Dla przykładowej sytuacji, przedstawionej na rysunku 7, półparabole przecinają się w jednym punkcie

Jak widać, dla tego konkretnego przykładu istnieje dokładnie jedno rozwiązanie. W ogólności układ równań (14) dopuszcza od 0 do 4 rozwiązań, gdyż jest to problem równoważny znajdowaniu miejsc zerowych wielomianu 4 stopnia i w zasadzie może być rozwiązany analitycznie. Jednakże znajdowanie punktu (lub punktów) przecięcia dwóch segmentów Béziera jest dość typowym zadaniem, nietrudnym do numerycznego rozwiązania. Wolno oczekiwać, że wszystkie systemy graficzne posługujące się krzywymi Béziera dostarczają wygodnych narzędzi do znajdowania punktu przecięcia dwóch łuków Béziera.

Takimi systemami są na przykład METAFONT Donalda E. Knutha i METAPOST Johna D. Hobby'ego (stanowiący modyfikację METAFONT-a). Obydwa programy wyposażone są w operację $\mathbf{P}_1(t)$ *intersectiontimes* $\mathbf{P}_2(t)$, zaimplementowane z wykorzystaniem prostego algorytmu bisekcji, która dla dowolnej pary krzywych Béziera $\mathbf{P}_1(t)$ i $\mathbf{P}_2(t)$ oblicza parę liczb rzeczywistych (t_1, t_2) ; jeżeli krzywe się nie przecinają, to $t_1 = t_2 = -1$, w przeciwnym wypadku $\mathbf{P}_1(t_1) = \mathbf{P}_2(t_2)$ (z dokładnością do precyzji obliczeń).

Jednakże operacja *intersectiontimes* nie dostarcza informacji o liczbie punktów przecięcia. W praktyce (zwłaszcza w zastosowaniach interakcyjnych) jednoznaczność rozwiązania nie jest kluczowa. W pierwszym przybliżeniu można założyć, że taka informacja, jakiej dostarcza operacja *intersectiontimes*, wystarcza. Oprogramowanie dostępu do wszystkich punktów przecięcia nie jest szczególnie trudnym zadaniem.

W przypadku przecinających się półparabol o prostopadłych osiach symetrii mogą się pojawić 1, 2 lub 3 punkty przecięcia (zob. rysunek 10). Można udowodnić, że nie może być 4 punktów przecięcia.



Rys. 10. Możliwe przecięcia dwóch półparabol o osiach wzajemnie prostopadłych; nie jest jasne, czy wszystkie przypadki są istotne w praktyce

ФАКТ: Dwie półparabole o prostopadłych osiach symetrii mogą przecinać się w co najwyżej 3 punktach.

ДОВОД: Bez straty ogólności możemy założyć, że osie półparabol pokrywają się z osiami układu współrzędnych. Przyjmijmy, że parabole zadane są równaniami $q = Q + Pp^2$ i $p = S + Rq^2$; półparabole odpowiadają dodatnim lub ujemnym wartościom argumentów p i q . Te dwa równania można potraktować jako układ 2 równań z 2 niewiadomymi p i q ; po wyeliminowaniu jednej zmiennej otrzymujemy równanie 4 stopnia $r^4 + \rho_3 r^3 + \rho_2 r^2 + \rho_1 r + \rho_0 = 0$ (gdzie r może oznaczać p lub q).

W przypadku półparabol o prostopadłych osiach zawsze zachodzi $\rho_3 = 0$, zaś dla wszystkich równań wielomianowych zachodzi, zgodnie ze wzorami Viète'a, $\rho_3 = r_1 + r_2 + r_3 + r_4$, gdzie r_1, r_2, r_3 i r_4 są pierwiastkami (rzeczywistymi lub zespolonymi) równania czwartego stopnia.

Jeżeli półparabole przecinają się w 3 punktach, to wynika stąd, że istnieją trzy pierwiastki rzeczywiste, powiedzmy r_1, r_2 i r_3 , mające ten sam znak. Zatem czwarty pierwiastek jest również rzeczywisty i ma znak przeciwny $r_4 = -(r_1 + r_2 + r_3)$. Oznacza to, że punkt przecięcia odpowiadający temu pierwiastkowi należy do drugiej gałęzi paraboli – c. b. d. o.

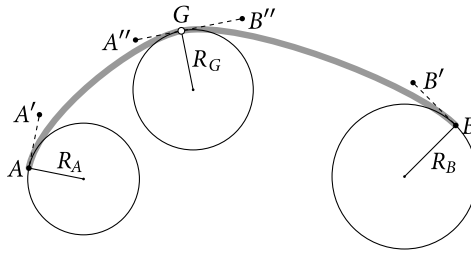
Przedstawiliśmy w zarysie implementację kluczowego modułu rozwiązującego problem promienia krzywizny. Niech moduł ten nazywa się przykładowo `find_controls(A, B, C, RA, RB)`. Jego specyfikacja jest następująca: dla danych punktów na płaszczyźnie A, B, C i liczb R_A, R_B albo obliczane jest położenie naciągów zgodnie ze sformułowaniem zadania, albo emitowany jest komunikat, że rozwiązanie nie istnieje. Jeśli dany system nie oferuje operacji mającej zbliżoną funkcjonalność do opisanej wyżej operacji `intersectiontimes`, niezbędnej dla zaimplementowania modułu `find_controls(A, B, C, RA, RB)`, można tego rodzaju operację względnie łatwo zaprogramować, stosując na przykład metodę bisekcji.

4.4. Możliwe zastosowania praktyczne

W punkcie 3 pokazaliśmy, że zawsze (z wyjątkiem przypadków zdegenerowanych), istnieje rozwiązanie następującego zadania:

PROBLEM GŁADKIEGO POŁĄCZENIA: Dla zadanych 6 punktów na płaszczyźnie A, A', A'', B'', B' i B , wyznaczyć punkt G , znajdujący się na linii prostej przechodzącej przez A'' i B'' , taki, że punkcie G krzywizna lewostronna równa jest prawostronnej krzywej zbudowanej z segmentów Béziera opartych na czworokątach $AA'A''G$ i $GB''B'B$.

Rozwiązanie tego zadania ilustruje rysunek 11. Jest to w zasadzie modyfikacja rysunku 3 – zaznaczone zostały, istotne dla dalszych rozważań, krzywizny w punktach A, G i B .



Rys. 11. Wyznaczanie punktu G łączącego segmenty Béziera z zachowaniem krzywizny; punkty A, A', A'', B'', B' i B są dane, promienie krzywizny R_A, R_B i R_G

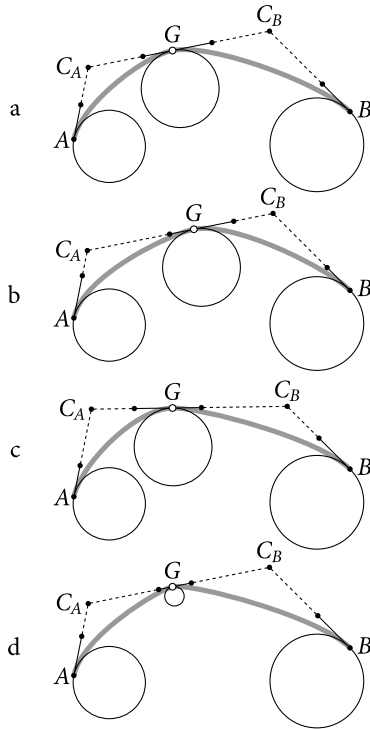
Wspomniane w punkcie 3.3 eksperymenty z implementacją algorytmu wyznaczania punktu G pokazały, że czasem użytkownik chciałby nieco zmodyfikować efekt działania algorytmu w otoczeniu punktu G . Będziemy zakładać, że położenia punktów A i B i kierunki naciągów w tych punktach nie podlegają modyfikacjom.

Jest kilka możliwości: można przesunąć punkt G (z zachowaniem lub nie krzywizny i kierunku stycznej w punkcie G); można nie zmieniać położenia punktu G , natomiast przesunąć stosownie A'' i B'' , co w istocie określa krzywiznę w punkcie G (należy ustalić, czy bierzemy pod uwagę lewo- czy prawostronną krzywiznę); wreszcie można zadać jawnie krzywiznę (i ewentualnie kierunek stycznej) w punkcie G . Do pomyślenia są oczywiście dowolne kombinacje wymienionych zabiegów.

Po dokonaniu zmian (w założeniu niewielkich) należy po prostu zastosować opisany ogólnie w punkcie 4.3 moduł *find_controls* z odpowiednimi parametrami: *find_controls*(A, C_A, G, R_A, R_G^l) oraz *find_controls*(G, C_B, B, R_G^p, R_B), gdzie C_A jest punktem przecięcia prostych przechodzących przez pary punktów AA' i $A''G$, C_B – punktem przecięcia prostych przechodzących przez BB' i $B''G$, zaś R_G^l i R_G^p oznaczają odpowiednio lewo- i prawostronną krzywiznę w punkcie G . W typowym przypadku $R_G^l = R_G^p$, nie jest to jednak warunek konieczny).

Jeżeli zmiany istotnie są niewielkie, można się spodziewać, że omawiany moduł znajdzie rozwiązanie dla zmodyfikowanej sytuacji. Rysunek 12 przedstawia wynik obliczeń dla trzech prostych modyfikacji.

Zauważmy, że skrajne naciągi przesuwają się nieznaczająco. To sugeruje następujące uproszczenie procedury modyfikującej otoczenie punktu G : założymy, że nie tylko położenie punktów A i B nie ulega zmianie, ale również odpowiednie naciągi



Rys. 12. (a) sytuacja początkowa; (b) zmiana położenia punktu G ; (c) zmiana kierunku w punkcie G ; (d) zmiana krzywizny w punkcie G

(w „gwarze” METAFONT-owej odpowiednio *postcontrol of A* i *precontrol of B*). Dzięki temu założeniu zadanie staje się trywialne.

Z równania (12) wynika, że jeżeli chcemy przesunąć jeden z naciągów, powiedzmy A' , zachowując krzywiznę w węzle A , nie zmieniając położenia węzłów i drugiego naciągu, to znaczy punktów A, B, B' , to po prostu wyznaczamy odległość H_A (według oznaczeń przyjętych na rysunku 7) i następnie obliczamy $|AA'|$ z zależności $R_A = \frac{3}{2} \frac{|AA'|^2}{H_A}$. Podobnie postępujemy, gdy chcemy przesunąć naciąg B' , zachowując krzywiznę w węzle B i nie zmieniając położenia punktów A, B, A' – wyznaczamy H_B i korzystamy z zależności $R_B = \frac{3}{2} \frac{|BB'|^2}{H_B}$.

5. Wnioski

Jak zostało wspomniane, eksperymentalna implementacja w edytorze fontów FONTLAB algorytmu opisanego w punkcie 3 przyniosła obiecujące rezultaty. Prace i próby trwają. Wolno żywić nadzieję, że udoskonalenie implementacji poprzez uwzględnienie opisanych w punkcie 4 technik modyfikowania otoczenia punktu G z zachowaniem ciągłości krzywizny w punkcie G i w sąsiednich węzłach przyczyni się do dalszego zwiększenia przydatności omawianego podejścia w praktyce.

Cytowana literatura

1. Hobby John. D., *Smooth, Easy to Compute Interpolating Splines*. „Discrete and Computational Geometry” 1986 vol. 1(2).
<http://i.stanford.edu/pub/cstr/reports/cs/tr/85/1047/CS-TR-85-1047.pdf> [dostęp 28.10.2014].
2. Knuth Donald E., *The METAFONTbook, Computers & Typesetting vol. C*. Addison-Wesley, Reading Massachusetts 1986.
3. Knuth Donald E., *The T_EXbook, Computers & Typesetting vol. A*. Addison-Wesley, Reading Massachusetts 1986.
4. Jackowski Bogusław, *Typografowie, programiści i matematycy, czyli przypadek estetycznie zadowolającej interpolacji*. „Acta Poligraphica” 2013 (1), s. 11–30.
<http://www.cobrrp.com.pl/actapoligraphica/uploads/pdf/AP2013.01.Jackowski.pdf>
5. Raphael Linus Levien, *From Spiral to Spline: Optimal Techniques in Interactive Curve Design*, praca doktorska, 2009.
<http://www.levien.com/phd/thesis.pdf> [dostęp 17.10.2014].
6. Manning J. R., *Continuity Conditions for Spline Curves*. „Computer Journal” 1974 vol. 17(2), s. 181–186.
<http://comjnl.oxfordjournals.org/content/17/2/181.full.pdf> [dostęp 27.10.2014].
7. *Adobe Type 1 Font Format*.
<http://partners.adobe.com/public/developer/en/font/T1.SPEC.PDF> [dostęp 28.10.2014].
8. *OpenType specification (full)*.
<http://download.microsoft.com/download/E/5/B/E5B69A0B-D636-46C1-8E08-97AE4CE09CAF/otspec160.zip> [dostęp 27.10.2014].
9. *Krzywa B-sklejana*.
http://pl.wikipedia.org/wiki/Krzywa_B-sklejana [dostęp 17.10.2014].
10. *Krzywa Béziera*.
http://pl.wikipedia.org/wiki/Krzywa_B%C3%A9ziera [dostęp 17.10.2014].
11. *Wielomiany Bernsteina*.
http://pl.wikipedia.org/wiki/Wielomiany_Bernsteina [dostęp 17.10.2014].

Abstract

On joining Bézier curves smoothly

The article presents an elementary formula for joining Bézier curves smoothly, i.e., keeping the the same curvature on both sides of a join, and a technique of modifying the neighbourhood of the joining point. The approach is not computationally demanding, so one can expect that it can be easily incorporated into any graphics system handling Bézier curves. An experimental implementation of the joining formula in a commercial font editor seems fairly promising.