

## » Idź do

- Spis treści
- Przykładowy rozdział

## » Katalog książek

- Katalog online
- Zamów drukowany katalog

## » Twój koszyk

- Dodaj do koszyka

## » Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

## » Czytelnia

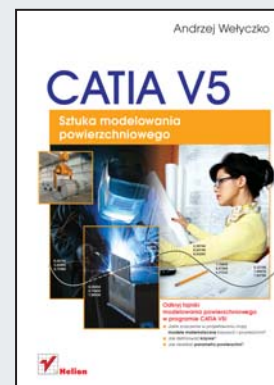
- Fragmenty książek online

## » Kontakt

Helion SA  
ul. Kościuszki 1c  
44-100 Gliwice  
tel. 032 230 98 63  
e-mail: helion@helion.pl  
© Helion 1991-2008

## CATIA V5. Sztuka modelowania powierzchniowego

Andrzej Wełyczko  
ISBN: 978-83-246-2393-8  
Format: 168×237, stron: 744



### Odkryj tajniki modelowania powierzchniowego w programie CATIA V5!

- Jakie znaczenie w projektowaniu mają modele matematyczne krzywych i powierzchni?
- Jak definiować krzywe?
- Jak określać parametry powierzchni?

Bez systemów CAD nikt nie wyobraża sobie dzisiaj profesjonalnego projektowania. Usprawniają one nie tylko sam proces projektowy, ale także tworzenie dokumentacji, oraz zapewniają łatwe i efektywne wprowadzanie zmian konstrukcyjnych. Jednak nie wszystkie systemy typu CAD oferują identyczne możliwości, szczególnie w kwestii modelowania powierzchniowego. Jeśli uważasz je za niezbędne w Twojej pracy, wypróbuj sprawdzony program CATIA V5 i naucz się optymalnie wykorzystywać jego zalety w tym zakresie.

Bogato ilustrowana książka „CATIA V5. Sztuka modelowania powierzchniowego” pozwoli Ci osiągnąć biegłość i całkowitą swobodę w dziedzinie modelowania. Zdradzi Ci wszelkie sekrety, pomagające zachować pełną kontrolę nad procesem zmian. Podpowie, jak używać różnych rodzajów krzywych, a także definiować kontury oraz relacje geometryczne i wymiarowe. Wyjaśni, jak zachować ciągłość modelu powierzchniowego i na czym polega praca w środowisku Generative Shape Design (GSD). Podsunie odpowiednie parametry i polecenia, a na dodatek wskaże mechanizmy wspomagające proces projektowania.

- Modele matematyczne krzywych i powierzchni
- Krzywe swobodne, Béziera, typu Spline i NURBS
- Moduł Sketcher – wybór płaszczyzny szkicowania, parametry konturu
- Środowisko Generative Shape Design – definiowanie krzywych
- Organizacja struktury modelu powierzchniowego
- Mechanizmy wspomagające projektowanie przestrzenne
- Określanie parametrów modelu powierzchniowego w GSD
- Definicja prostych powierzchni parametrycznych
- Definicja powierzchni wymagających krzywej typu Spine
- Zapewnianie wymaganej ciągłości modelu powierzchniowego
- Operacje na powierzchniach teoretycznych
- Analizy jakości powierzchni

**Wszystkie sekrety efektywnego modelowania powierzchniowego w CATIA V5!**

# Spis treści

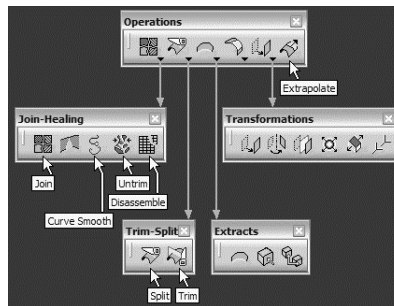
<b>Wstęp</b> .....	<b>5</b>
<b>Rozdział 1. Modele matematyczne krzywych i powierzchni</b> .....	<b>13</b>
Wprowadzenie .....	13
Ciągłość geometryczna .....	14
Gładkość .....	18
Model matematyczny krzywej lub powierzchni .....	20
Krzywe .....	24
Krzywe swobodne typu Natural Spline .....	25
Krzywe Béziera .....	26
Krzywe typu Spline .....	38
Krzywe B-Spline .....	43
Krzywe NURBS .....	56
Powierzchnie .....	62
<b>Rozdział 2. Definiowanie krzywych</b> .....	<b>69</b>
Sketcher .....	69
Wybór płaszczyzny szkicowania .....	70
Lokalny układ współrzędnych konturu — pozycjonowanie konturu .....	80
Definiowanie geometrii konturu .....	89
Definiowanie relacji geometrycznych i wymiarowych .....	129
Analizy konturu .....	134
Kopiowanie konturów .....	142
Generative Shape Design .....	147
Definiowanie elementów podstawowych .....	148
Typowe krzywe parametryczne .....	151
Krzywe definiowane na podstawie innych krzywych .....	160
Krzywe definiowane w powiązaniu z powierzchniami .....	199
Definiowanie krzywych konstrukcyjnych .....	222
Analizy krzywych .....	253
Organizacja struktury modelu powierzchniowego .....	265
Mechanizmy wspomagające projektowanie przestrzenne .....	280
<b>Rozdział 3. Definiowanie powierzchni</b> .....	<b>309</b>
Proces definiowania modelu powierzchniowego w środowisku GSD .....	309
Definicja prostych powierzchni parametrycznych .....	311
Polecenie Extrude .....	311
Polecenie Revolve .....	313
Polecenia Sphere i Cylinder .....	315

Definicja powierzchni wymagających krzywej typu Spine .....	320
Polecenie Multi-sections Surface .....	325
Polecenie Sweep .....	364
Polecenie Law .....	448
Polecenie Adaptive Sweep .....	465
Zapewnienie wymaganej ciągłości modelu powierzchniowego .....	491
Polecenie Fill .....	493
Grupa poleceń Fillets .....	502
Polecenie Blend .....	563
Operacje na powierzchniach teoretycznych .....	598
Polecenie Extrapolate .....	598
Polecenia Split i Untrim .....	614
Polecenia Join, Healing i Trim .....	627
Polecenia Disassemble, Extract i Multiple Extract .....	643
Grupa poleceń OffsetVar .....	646
Analizy jakości powierzchni .....	656
Polecenie Connect Checker .....	659
Polecenie Surface Curvature Analysis .....	664
Polecenie Feature Draft Analysis .....	680
<b>Zakończenie .....</b>	<b>719</b>
<b>CD-ROM .....</b>	<b>721</b>
<b>Skorowidz .....</b>	<b>725</b>

## Definiowanie krzywych konstrukcyjnych

Krzywe konstrukcyjne to takie krzywe, których definicja nie zawsze jest możliwa za pomocą pojedynczych poleceń środowiska *Generative Shape Design*. Definiowanie krzywych elementarnych nie zawsze spełnia wszystkie wymagania konstrukcyjne, bo konstruktor musi czasami przyciąć zbyt długie krzywe, podzielić krzywą na kilka części lub wykonać jakąś transformację geometryczną. Polecenia wspomagające konstruktora w tego typu zadaniach znajdują się w pasku narzędziowym *Operations* (rysunek 2.269). Niektóre z tych narzędzi omówiono już wcześniej (grupa *Extracts*: polecenia *Boundary* i *Extract*), a część z nich (polecenie *Extrapolate* oraz grupę *Transformations*) pozostawię bez komentarza, bo moim zdaniem transformacje geometryczne typu obrót lub symetria, lub ekstrapolacja krzywej są po prostu trywialne.

**Rysunek 2.269.**  
Zestawienie poleceń  
grupy *Operations*  
środowiska *Generative  
Shape Design*



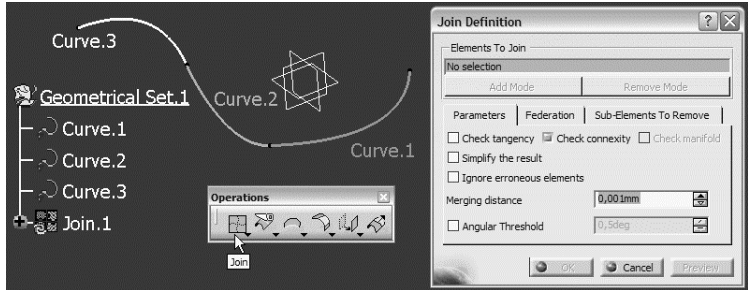
## Polecenie Join

Polecenie *Join* służy do łączenia kilku krzywych lub powierzchni w jeden obiekt geometryczny. Po co łączyć krzywe? Połączenie wielu drobnych krzywych jest często jedynym możliwym rozwiązaniem, zwłaszcza wtedy, gdy krzywe podstawowe modelu powierzchniowego (zadane przekroje powierzchni, krzywe prowadzące lub krzywa typu *Spine*) nie

mogą być zdefiniowane za pomocą jednego polecenia. Na pierwszy rzut oka działanie tego polecenia jest bardzo intuicyjne. Wystarczy zaznaczyć wszystkie krzywe, które mają być połączone (rysunek 2.270 i 2.271), czyli wskazać każdą krzywą lub zastosować dowolny tryb wyboru wielokrotnego (*Multi-Select* lub *Search*) i zaakceptować definicję krzywej typu *Join* (rysunek 2.272).

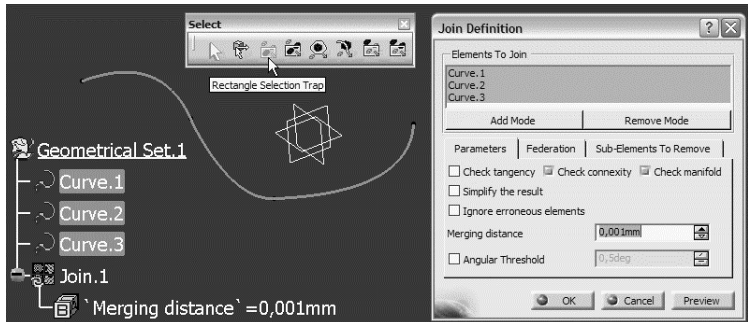
**Rysunek 2.270.**

Zastosowanie polecenia  
*Join* — krok 1.



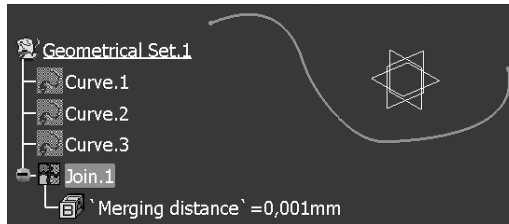
**Rysunek 2.271.**

Zastosowanie polecenia  
*Join* — krok 2.



**Rysunek 2.272.**

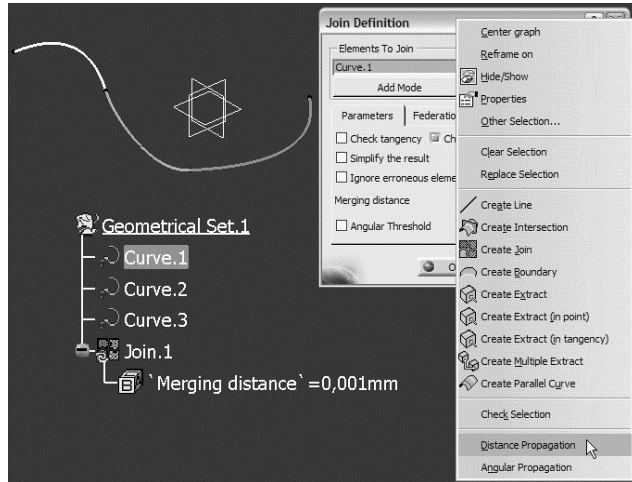
Zastosowanie polecenia  
*Join* — krok 3.



Krzywe elementarne definiowane przez konstruktora są zawsze topologicznie poprawne. Nieco trudniej zapewnić taką jakość krzywych, które powstały na przykład w wyniku przecięcia dwóch powierzchni lub rzutowania krzywej na powierzchnię. W takich przypadkach po uruchomieniu polecenia *Join* dobrze jest włączyć opcję *Check manifold*. Jeśli łączymy kilka krzywych, to zazwyczaj nieświadomie zakładamy, że wszystkie te krzywe są wzajemnie ciągłe, przynajmniej według kryterium G0. Łączenie krzywych, które się wzajemnie przecinają, lub takich, których punkty skrajne nie są wspólne (w zakresie pewnej tolerancji), nie ma przecież praktycznego uzasadnienia w projektowaniu powierzchniowym. W tym sensie wybór krzywych do połączenia może wspomagać polecenie *Distance Propagation*, dostępne po wskazaniu przynajmniej jednej krzywej w menu kontekstowym obszaru *Elements To Join* okna *Join Definition* (rysunek 2.273).

**Rysunek 2.273.**

Wybór krzywych  
za pomocą polecenia  
*Distance Propagation*

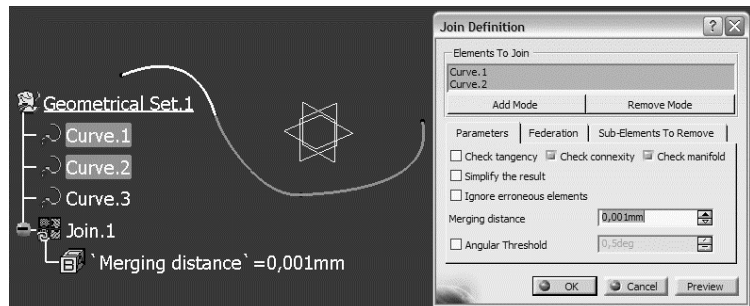


W rezultacie zastosowania polecenia *Distance Propagation* zostaną wybrane wszystkie krzywe, które są wzajemnie ciągłe. Wartość parametru *Merging distance* określa w tym przypadku lokalną tolerancję identyczności punktów stosowaną przez algorytm polecenia *Join*:

- ◆ Jeśli odległość dwóch punktów skrajnych kolejnych (sąsiadujących) krzywych jest mniejsza od *Merging distance*, to system traktuje te dwa punkty jako identyczne. Nie jest możliwe zdefiniowanie odcinka linii prostej, którego punkty skrajne są odległe o mniej niż *Merging distance*.
- ◆ Jeśli odległość dwóch punktów skrajnych kolejnych (sąsiadujących) krzywych jest większa od *Merging distance*, to system traktuje te dwa punkty jako różne i dlatego można zdefiniować odcinek linii prostej lub dowolną krzywą łączącą te punkty. Automatyczny wybór krzywych (*Distance Propagation*) jest w takim przypadku zatrzymywany. Na przykład jeśli krzywe *Curve.2* i *Curve.3* nie są ciągłe, a odległość ich punktów skrajnych jest większa od standardowej wartości *Merging distance = 0,001 mm*, to po wskazaniu krzywej *Curve.1* i uruchomieniu polecenia *Distance Propagation* propagacja wyboru zatrzyma się na krzywej *Curve.2* (rysunek 2.274).

**Rysunek 2.274.**

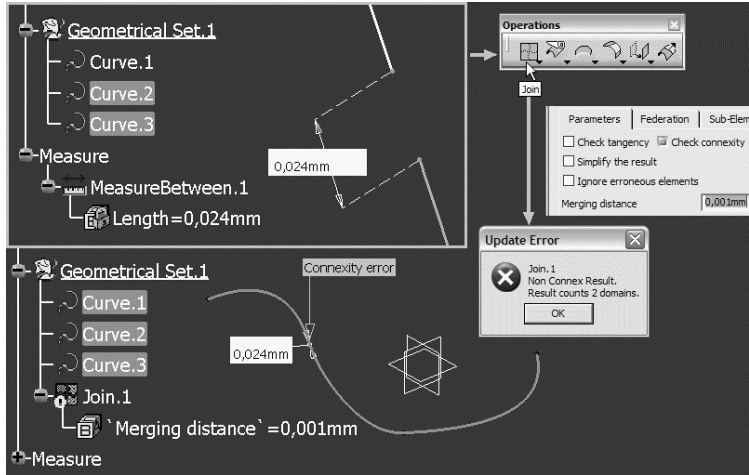
Rezultat zastosowania  
polecenia *Distance  
Propagation*



Możliwe jest „ręczne” wskazanie krzywej *Curve.3* i zatwierdzenie wykonania polecenia *Join*, ale przy włączonej opcji *Check connectivity* pojawia się błąd wykonania, bo wynikowa krzywa *Join.1* składa się z dwóch nieciągłych części (rysunek 2.275).

**Rysunek 2.275.**

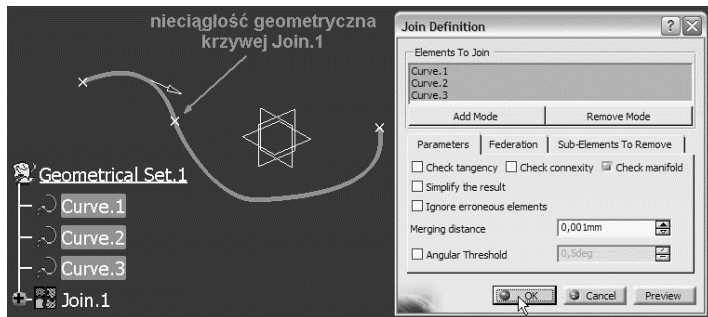
Przykład błędu wykonania polecenia *Join* dla krzywych bez ciągłości geometrycznej



Wydaje się, że najprostszym rozwiązaniem jest wyłączenie opcji *Check connectivity*, czyli trybu sprawdzania ciągłości łączonych krzywych. Nie wpływa to jednak w żaden sposób na jakość krzywej *Join.1*, bo mimo że jej definicja jest możliwa dla krzywych nieciągłych, to brak ciągłości krzywej wynikowej jest sygnalizowany znakami X. Topologicznie krzywa *Join.1* jest więc jednym obiektem (rysunek 2.276), ale z powodu nieciągłości jej krzywych składowych, także ich „suma” nie ma ciągłości geometrycznej.

**Rysunek 2.276.**

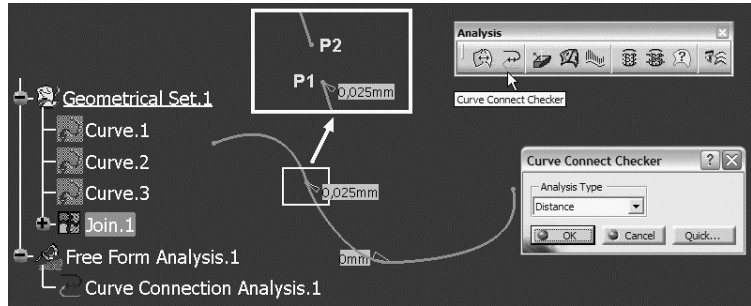
Przykład krzywej typu *Join* z nieciągłością geometryczną



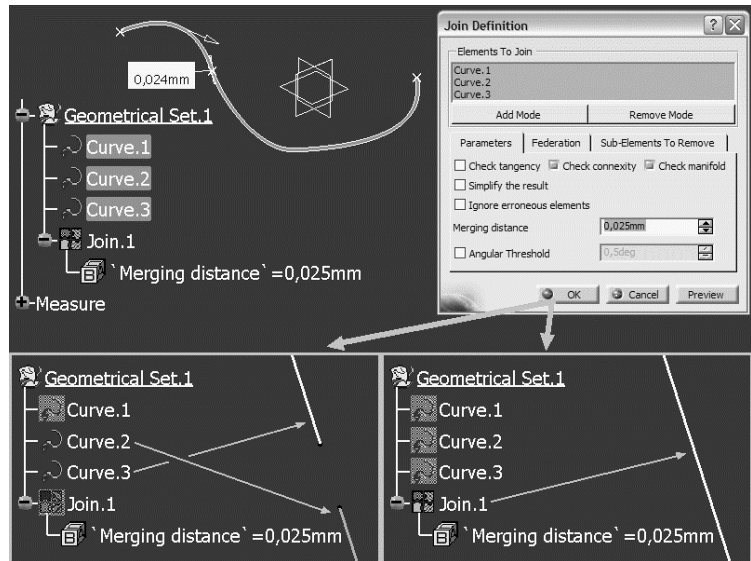
Brak ciągłości geometrycznej krzywej *Join.1* oraz wielkość tej nieciągłości można łatwo sprawdzić za pomocą polecenia *Curve Connect Checker* (rysunek 2.277). Rezultat tej analizy wyraźnie dowodzi, że polecenie *Join* nie naprawia geometrii (nie modyfikuje krzywych wejściowych), a jedynie łączy te krzywe w jeden logicznie spójny obiekt geometryczny. Punkty *P1* i *P2* są topologicznie identyczne, chociaż geometrycznie są to dwa różne punkty.

Gdybyśmy zwiększyli tolerancję do wartości większej od zmierzonej nieciągłości krzywych *Curve.2* i *Curve.3*, na przykład *Merging distance* = 0,025 mm, to polecenie *Distance Propagation* zakończy się wyborem wszystkich trzech krzywych, a krzywa *Join.1* wygląda na ciągłą według kryterium G0 (rysunek 2.278). Wygląda na ciągłą, ale taką nie jest, bo wartość parametru *Merging distance* nie ma żadnego wpływu na ciągłość krzywej *Join.1*.

**Rysunek 2.277.**  
*Analiza ciągłości geometrycznej krzywej Join.1*



**Rysunek 2.278.**  
*Definicja krzywej Join.1 z tolerancją ciągłości Merging distance = 0,025 mm*



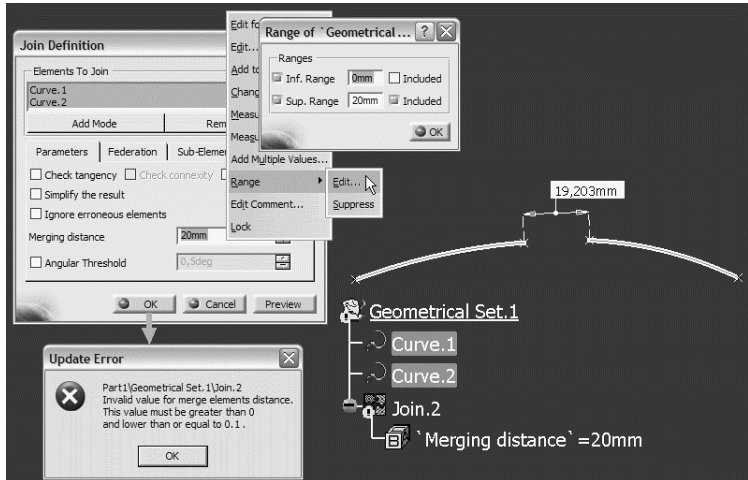
Wartość parametru *Merging distance* powinna być większa od zmierzonej nieciągłości krzywych. Nie zaleca się stosowania wartości *Merging distance* = *Measure Between*.

Wartość parametru *Merging distance* nie może być większa od  $0,1 \text{ mm}$ . Nie można przecież oczekiwać, że system będzie poprawiał każdą niedoróbkę lub wyręczał leniwego konstruktora. Mimo że teoretycznie możliwa jest zmiana zakresu wartości tego parametru (rysunek 2.279), to algorytm polecenia *Join* nie akceptuje wartości większych niż  $0,1 \text{ mm}$  — próba wykonania tego polecenia z wartością *Merging distance*  $> 0,1 \text{ mm}$  kończy się błędem wykonania.

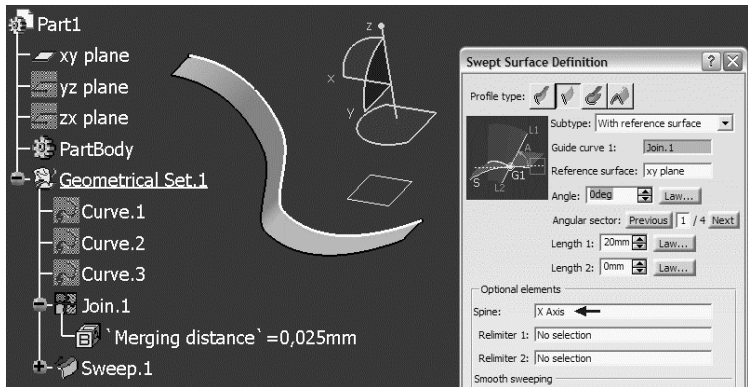
Krzywa *Join.1* zdefiniowana z maksymalną wartością parametru *Merging distance* mimo swej niedoskonałości może być w ograniczonym zakresie zastosowana do budowy modelu powierzchniowego. Na przykład powierzchnia *Sweep.1* (rysunek 2.280), zdefiniowana jako przeciągnięcie segmentu liniowego o długości  $Length1 = 20 \text{ mm}$  wzdłuż krzywej *Join.1* z zachowaniem kąta  $Angle = 0 \text{ deg}$  z powierzchnią bazową *Reference surface* =  $xy \text{ plane}$ , jest całkowicie poprawna. Jeśli jednak krzywa typu *Spine* tej powierzchni zostanie



**Rysunek 2.279.**  
Maksymalna wartość parametru *Merging distance* = 0,1 mm nie może być zmieniona



**Rysunek 2.280.**  
Przykład powierzchni typu *Sweep* z *Join.1* jako krzywą prowadzącą i *X Axis* jako *Spine*

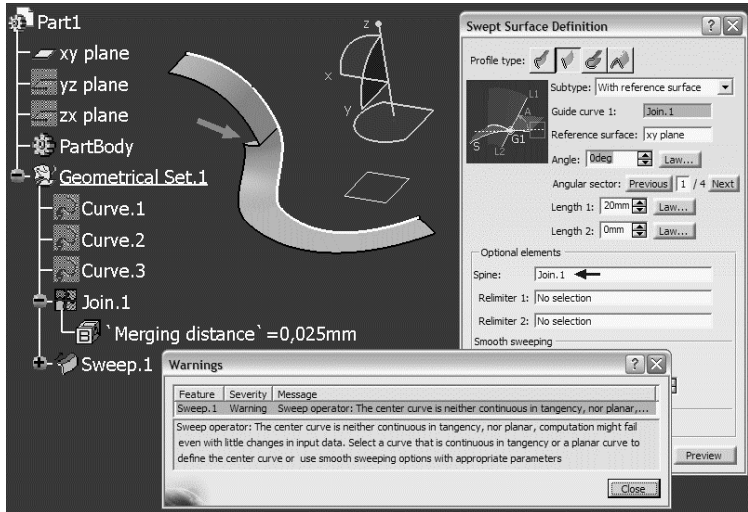


zmieniona na *Join.1* (rysunek 2.281), to system wyświetli komunikat z ostrzeżeniem o braku ciągłości stycznych (typ *G1*) tej krzywej. Definicja powierzchni *Sweep.1*, pomimo problematycznej definicji krzywej *Join.1*, jest możliwa, ale model geometryczny nie jest stabilny — niewielkie zmiany geometrii lub wartości parametrów mogą być powodem błędu wykonania polecenia *Sweep*.

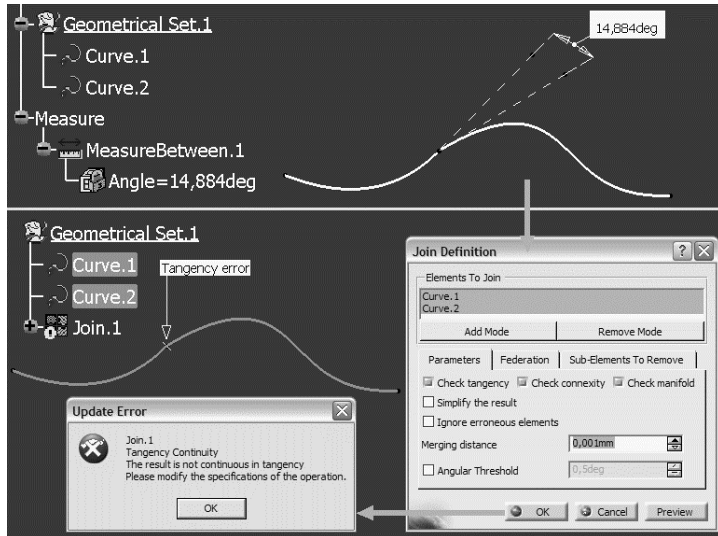
Powyższy przykład krzywej złożonej z kilku innych krzywych elementarnych powinien przypomnieć Czytelnikowi, że ciągłość krzywej to nie tylko ciągłość geometryczna (*G0*), bo w większości przypadków konstrukcyjnych wymagania stawiane krzywym to przynajmniej ciągłość stycznych (*G1*). Dlatego polecenie *Join* umożliwia analizę ciągłości stycznych krzywych wejściowych (aktywna opcja *Check tangency*). Na przykład jeśli dla krzywych *Curve.1* i *Curve.2* (rysunek 2.282) kąt pomiędzy stycznymi do obu krzywych w ich punkcie wspólnym ma wartość  $14,884 \text{ deg}$ , to próba połączenia tych krzywych z aktywną opcją *Check tangency* generuje błąd wykonania.

Gdyby dla bardzo podobnych krzywych kąt pomiędzy stycznymi w ich punkcie wspólnym był mniejszy od  $0,5 \text{ deg}$ , na przykład  $0,349 \text{ deg}$ , to zastosowanie polecenia *Join* do takich krzywych nie generuje żadnego błędu wykonania (rysunek 2.283), bo  $0,5 \text{ deg}$  jest standardową tolerancją styczności krzywych w systemie CATIA V5. Jeśli kąt stycznych do dwóch

**Rysunek 2.281.**  
Przykład powierzchni  
typu Sweep z Join.1  
jako krzywą prowadzącą  
i jednocześnie jako Spine



**Rysunek 2.282.**  
Przykład błędu  
wykonania polecenia  
Join dla krzywych  
bez ciągłości stycznych



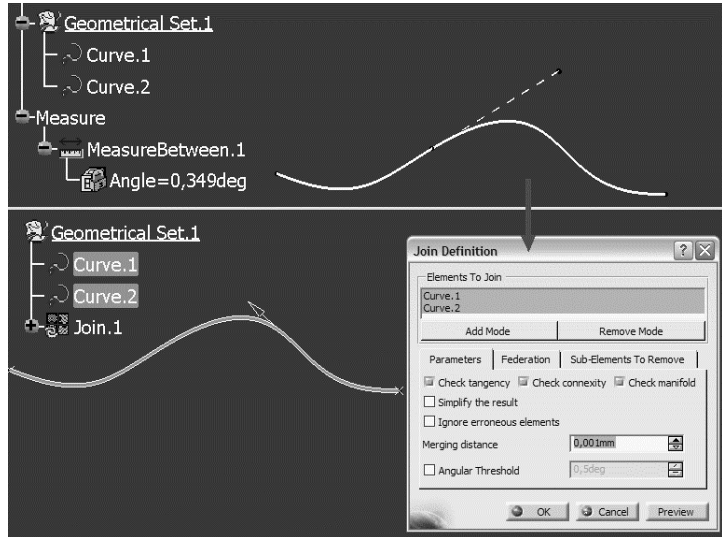
krzywych w ich punkcie wspólnym jest mniejszy od tolerancji styczności, to system traktuje te krzywe jako ciągłe według kryterium G1. Zwracam uwagę, że określenie „kął pomiędzy stycznymi do dwóch krzywych w ich punkcie wspólnym” oznacza ciągłość krzywych przynajmniej według kryterium G0 i dlatego aktywna opcja *Check tangency* wymusza aktywność opcji *Check connexity*.

Gdyby dla bardzo podobnych krzywych kął pomiędzy stycznymi w ich punkcie wspólnym był większy od  $0,5 \text{ deg}$ , na przykład  $0,999 \text{ deg}$ , to zastosowanie polecenia *Join* do takich krzywych generuje błąd wykonania (rysunek 2.284).

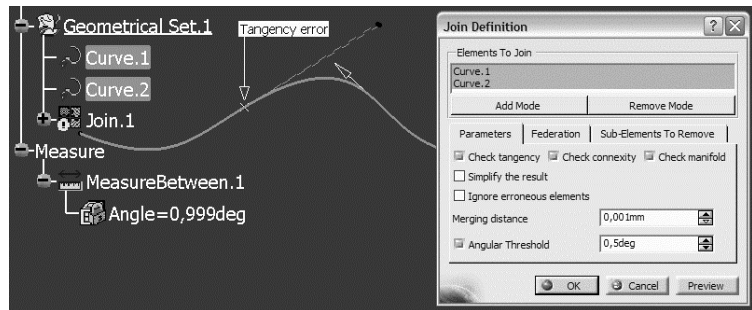
Większa wartość tolerancji styczności krzywych *Curve.1* i *Curve.2* w ich punkcie wspólnym, na przykład *Angular Threshold = 1,5 deg*, generuje topologicznie ciągłą krzywą *Join.1*

**Rysunek 2.283.**

Przykład krzywej typu  
Join z nieciągłością  
styczności

**Rysunek 2.284.**

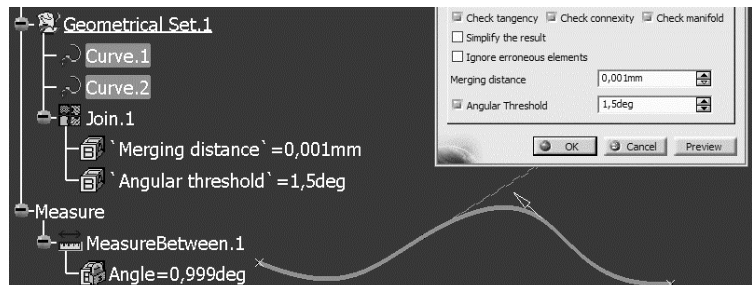
Przykład błędu  
wykonania polecenia  
Join dla krzywych  
bez ciągłości stycznych  
dla wartości parametru  
Angular Threshold  
mniejszej od kąta  
stycznych do krzywych  
Curve.1 i Curve.2



(rysunek 2.285). Krzywa *Join.1* wygląda na ciągłą według kryterium G1, ale taką nie jest, bo nadmierne zwiększanie wartości parametru *Angular Threshold* nie rozwiązuje problemu nieciągłości geometrycznej tej krzywej.

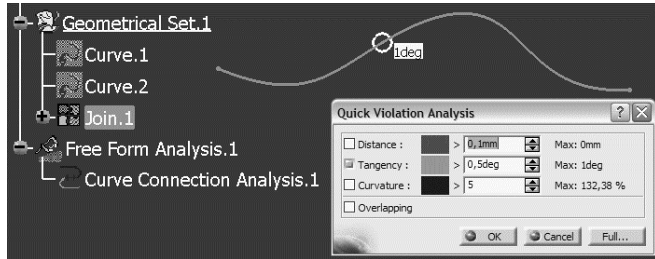
**Rysunek 2.285.**

Przykład krzywej typu  
Join z nieciągłością  
styczności



Brak ciągłości stycznych krzywej *Join.1* oraz wielkość tej nieciągłości można łatwo sprawdzić za pomocą polecenia *Curve Connect Checker* (rysunek 2.286). W rezultacie po raz kolejny widać, że polecenie *Join* nie naprawia geometrii (nie modyfikuje krzywych

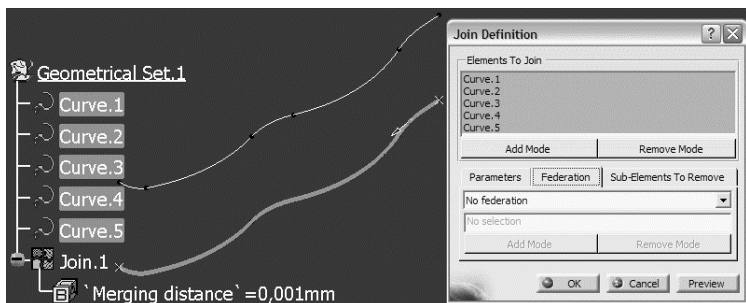
**Rysunek 2.286.**  
*Analiza ciągłości stycznych krzywej Join.1*



wejściowych), a jedynie łączy te krzywe w jeden logicznie spójny obiekt geometryczny — krzywa *Join.1* ma w punkcie wspólnym krzywych *Curve.1* i *Curve.2* nieciągłość stycznych o wartości *1 deg*.

Zakładka *Parameters* okna *Join Definition* zwykle wystarcza do definicji krzywej typu *Join*. Jednak czasami konstruktor potrzebuje trochę więcej niż tylko klasyczną definicję krzywej sklejanej i stosuje nieco inną definicję krzywej sklejanej dostępną w zakładce *Federation*. Co to jest *Federation*? Tu pozwolę sobie na pewną analogię geopolityczną. *Federacja* (ang. *federation*) to grupa państw posiadających pewną autonomię, ale bez granic wewnętrznych. *Konfederacja* (ang. *confederation*) jako pojęcie przeciwne federacji to także grupa państw, ale z wyraźnie wytyczonymi granicami wewnętrznymi. Klasyczna definicja krzywej typu *Join* jest odpowiednikiem konfederacji kilku krzywych. Pomimo „sklejenia” w jeden obiekt (rysunek 2.287) system ciągle rozpoznaje krzywe cząstkowe oraz ich punkty skrajne (rysunek 2.288). Wskazanie krzywej lub jej punktu granicznego generuje komunikat systemowy w lewym dolnym rogu ekranu (*Edge\Join.1\Geometrical Set.1* lub *Vertex\Join.1\Geometrical Set.1*). Tu chciałbym zauważyć, że krzywa sklejona może mieć swoją historię (listę krzywych cząstkowych) lub nie, ale nawet jeśli *Join.1* zdefiniowano jako element typu *Datum*, to taka krzywa też może „pamiętać”, z jakich segmentów została sklejona (*Edge\Curve.6\Geometrical Set.1* lub *Vertex\Curve.6\Geometrical Set.1*). O tym, czy zachowany jest dostęp do wewnętrznych krzywych cząstkowych, decyduje aktywność trybu *Federation* podczas definicji krzywej typu *Join*.

**Rysunek 2.287.**  
*Klasyczna definicja krzywej typu Join w trybie Federation = No federation*

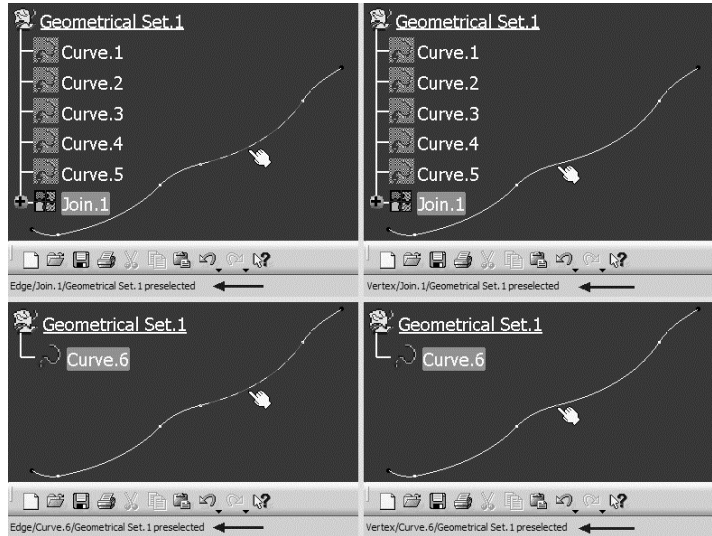


Jaki wpływ może mieć taka definicja krzywej sklejanej na kolejne etapy procesu projektowego? Załóżmy, że konstruktor potrzebuje krzywej równoległej do jednego z segmentów krzywej *Join.1*. Taką krzywą można zdefiniować bezpośrednio, odwołując się do odpowiedniej krzywej cząstkowej (*Curve.1* – *Curve.5*) lub pośrednio przez segment krzywej *Join.1*. Jeśli tylko definicja krzywej *Join.1* zapewnia dostęp do jej „wewnętrznych” Jeśli ta sama krzywa *Join.1* zostałaby zdefiniowana w trybie *Federation = Tangent*

segmentów (*Federation = No federation*), to można wskazać taki segment jako *Element(s)* to *extract* polecenia *Extract* (rysunek 2.289), a następnie zdefiniować krzywą równoległą (rysunek 2.290).

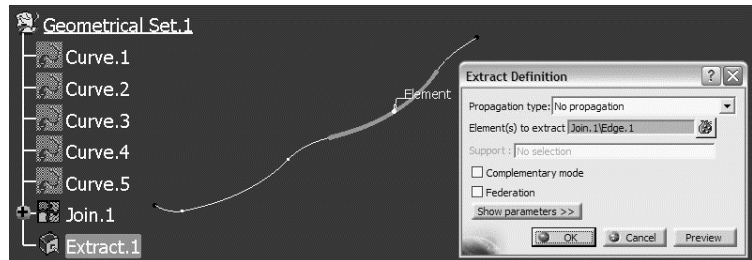
**Rysunek 2.288.**

Segmenty i granice pomiędzy tymi segmentami dostępne w definicji krzywej *Join* w trybie *Federation = No federation*



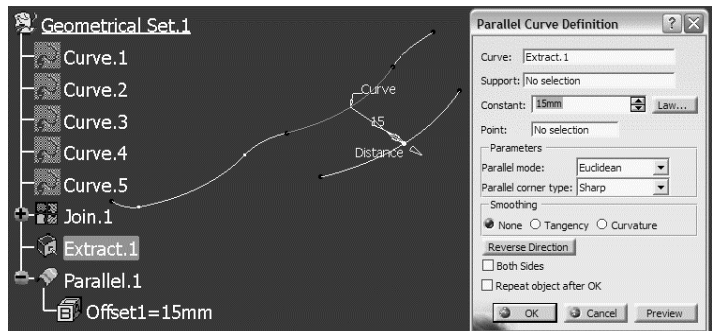
**Rysunek 2.289.**

Polecenie *Extract* zastosowane do segmentu krzywej *Join.1*



**Rysunek 2.290.**

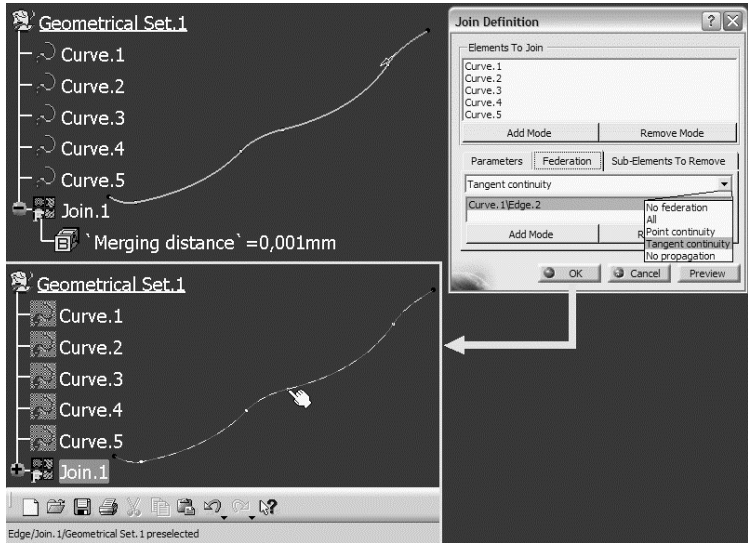
Definicja krzywej równoległej *Parallel.1*



*continuity* (także *Point continuity* lub *All*), to segmenty krzywej sklejonej nie będą rozpoznawane — wskazanie dowolnego z nich oznacza wskazanie całej krzywej *Join.1* (rysunek 2.291). Wynik wskazania krzywej kursorem jest więc taki sam jak wskazanie tej krzywej w drzewie strukturalnym modelu.

**Rysunek 2.291.**

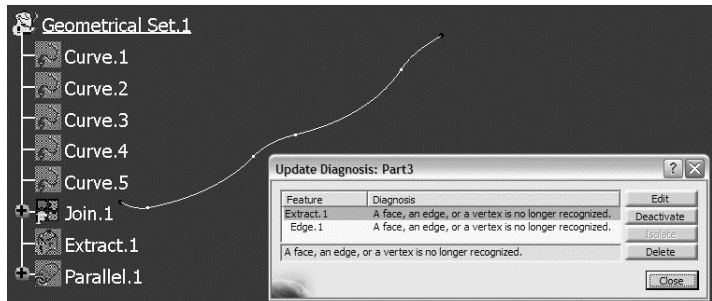
Definicja krzywej typu *Join* w trybie *Federation = Tangent continuity*



Gdybyśmy dla krzywej typu *Join*, która jest elementem nadrzędnym innych elementów geometrycznych (na przykład dla krzywej *Join.1* z rysunku 2.290), zmodyfikowali jej definicję przez włączenie trybu *Federation*, to w konsekwencji pojawi się błąd wykonania operacji *Update* — krzywa *Join.1* zdefiniowana w trybie *Federation = Tangent continuity* nie ma przecież segmentów i dlatego nie jest możliwa definicja krzywych *Extract.1* oraz *Parallel.1* (rysunek 2.292).

**Rysunek 2.292.**

Błąd wykonania operacji *Update* po zmianie trybu z *Federation = No federation* na *Federation = Tangent continuity*

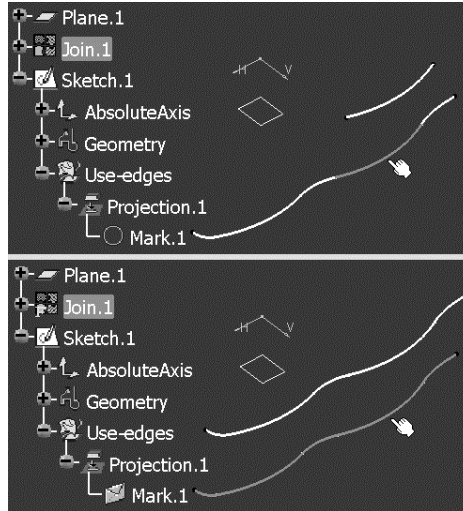


Kolejny przykład zastosowania krzywej sklejaney może być związany z definiowaniem konturów w środowisku *Sketcher*. Gdyby fragmentem konturu *Sketch.1* miał być rzut (*Projection.1*) wskazanego segmentu krzywej *Join.1* na płaszczyznę konturu (*Plane.1*), to w zależności od rodzaju definicji krzywej *Join.1* otrzymamy inny rezultat (rysunek 2.293):

- ◆ Klasyczna definicja krzywej *Join.1* — tylko wskazany segment krzywej jest rzutowany,
- ◆ Definicja krzywej *Join.1* z aktywnym trybem *Federation*:
  - ◆ Dla *Federation = All* — wszystkie segmenty krzywej *Join.1* niezależnie od rodzaju ciągłości ze wskazaniem segmentem (także nieciągłe!) będą rzutowane,

**Rysunek 2.293.**

Przykłady rzutowania na płaszczyznę konturu segmentu przestrzennej krzywej *Join.1* zdefiniowanej z wyłączonym i włączonym trybem *Federation*



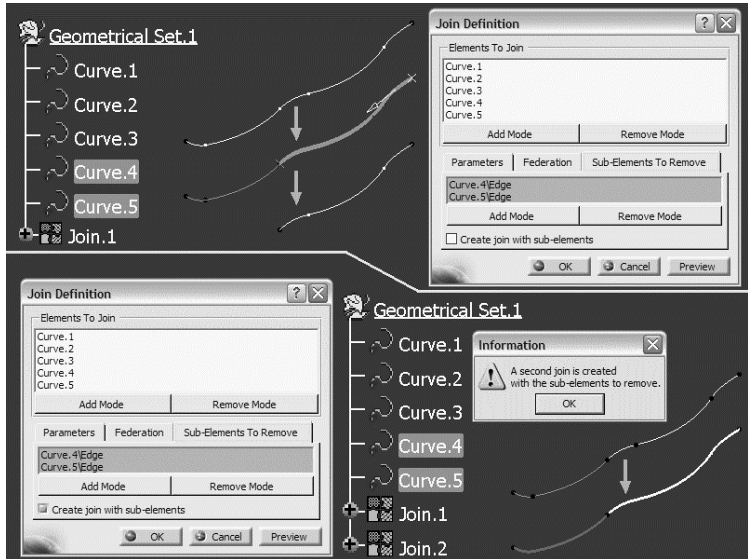
- ♦ Dla *Federation = Point continuity* — rzutowane będą tylko te segmenty krzywej *Join.1*, które są ciągłe według kryterium  $G0$  ze wskazanym segmentem,
- ♦ Dla *Federation = Tangent continuity* — rzutowane będą tylko te segmenty krzywej *Join.1*, które są ciągłe według kryterium  $G1$  ze wskazanym segmentem.

Ostatnia z zakładki okna *Join Definition*, czyli *Sub-Elements To Remove*, umożliwia zdefiniowanie podzbioru krzywych wskazanych w polu *Elements To Join*. Gdybyśmy dla tych samych krzywych *Curve.1*, *Curve.2* – *Curve.5* zdefiniowali krzywą *Join.1* i w polu *Sub-Elements To Remove* wskazali do usunięcia z definicji krzywej sklejanej krzywe *Curve.4* i *Curve.5*, to w rezultacie krzywa *Join.1* byłaby sklejona tylko z krzywymi *Curve.1*, *Curve.2* i *Curve.3* (rysunek 2.294). Taki sam rezultat można uzyskać bez definiowania listy elementów do usunięcia (*Sub-Elements To Remove*) przez ograniczenie listy *Elements To Join*. Jedyne praktyczne zastosowanie tej zakładki to możliwość definiowania dwóch krzywych sklejanych jednocześnie.

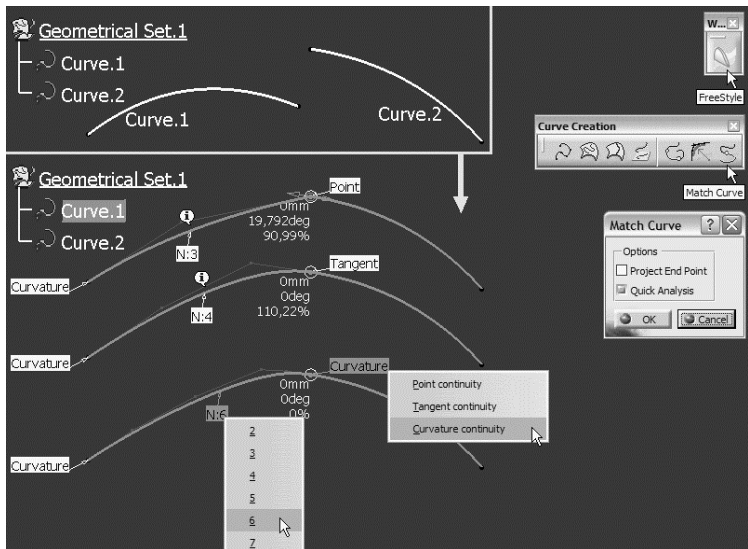
Przy okazji omawiania tematu krzywych sklejanych nie sposób pominąć zagadnień związanych z naprawą nieciągłości krzywych cząstkowych. Niestety, w środowisku *GSD* nie ma odpowiednika polecenia *Match Curve*, które jest dostępne w środowisku *FreeStyle*. Kształt krzywej *Curve.1* jest na rysunku 2.295 modyfikowany automatycznie w taki sposób, aby z krzywą *Curve.2* uzyskać wymagany rodzaj ciągłości (*Point*, *Tangency* lub *Curvature*). Inaczej mówiąc, kształt krzywej *Curve.1* wynika ze stopnia tej krzywej oraz warunków ciągłości zdefiniowanych w jej punktach skrajnych.

Brak polecenia typu *Match Curve* w środowisku *GSD* nie oznacza jednak, że konstruktor nie może naprawić nieciągłości krzywych. Musi to jednak zrobić „ręcznie”, stosując na przykład sekwencję poleceń: *Split*, potem *Connect Curve* lub *Spline* i na zakończenie *Join*.

**Rysunek 2.294.**  
Zastosowanie listy  
elementów do usunięcia  
(Sub-Elements  
To Remove) w definicji  
krzywej typu Join



**Rysunek 2.295.**  
Automatyczna  
modyfikacja kształtu  
krzywej w środowisku  
FreeStyle wymuszona  
przez warunki ciągłości  
i stopień krzywej



## Polecenie Split

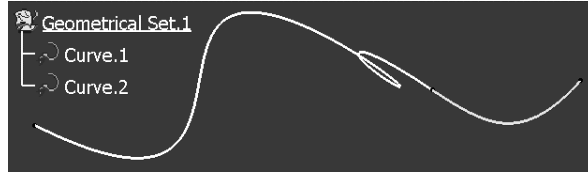
Jeśli krzywa jest wynikiem rzutowania lub przecięcia dwóch powierzchni, to może się zdarzyć, że jej kształt jest lokalnie niedopuszczalny — patrz krzywa z „pętlą” na rysunku 2.296. W takim przypadku problematyczny fragment krzywej może być odcięty i odrzucony (polecenie *Split* — rysunek 2.297), a następnie „uzupełniony” krzywą ciągłą, którą można zdefiniować za pomocą polecenia *Connect* lub *Spline* (rysunek 2.298). Jako element tnący można zastosować dowolny element z grupy *Wireframe*, czyli punkt, linię, krzywą, płaszczyznę lub powierzchnię.



W rezultacie po zastosowaniu polecenia *Join* uzyskamy poprawną geometrycznie i ciągłą krzywą *Join.1* (rysunek 2.299), a rodzaj ciągłości krzywej wynikowej zależy od parametrów ustalonych w definicji krzywej łączącej (*Spline* lub *Connect*).

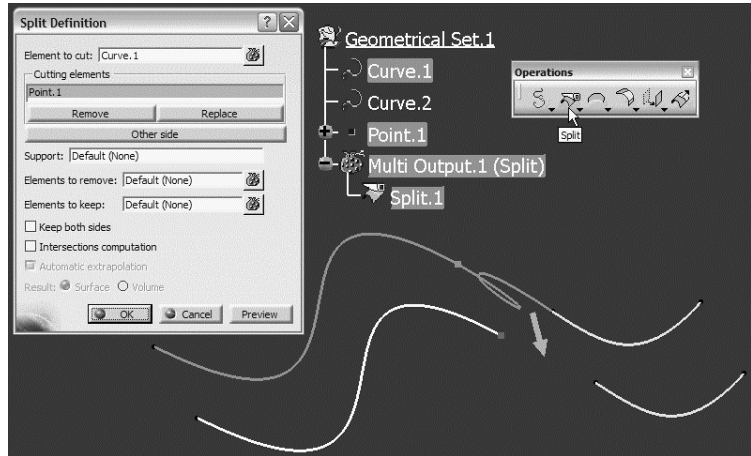
**Rysunek 2.296.**

Przykład krzywej z „pętlą”



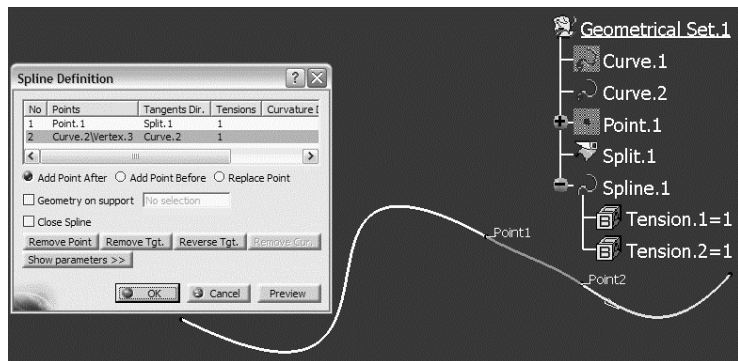
**Rysunek 2.297.**

Odcięcie fragmentu krzywej za pomocą polecenia *Split*



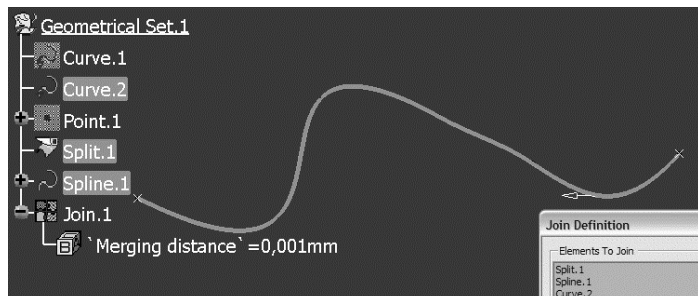
**Rysunek 2.298.**

Uzupełnienie odrzuconego fragmentu krzywej za pomocą polecenia *Spline*



**Rysunek 2.299.**

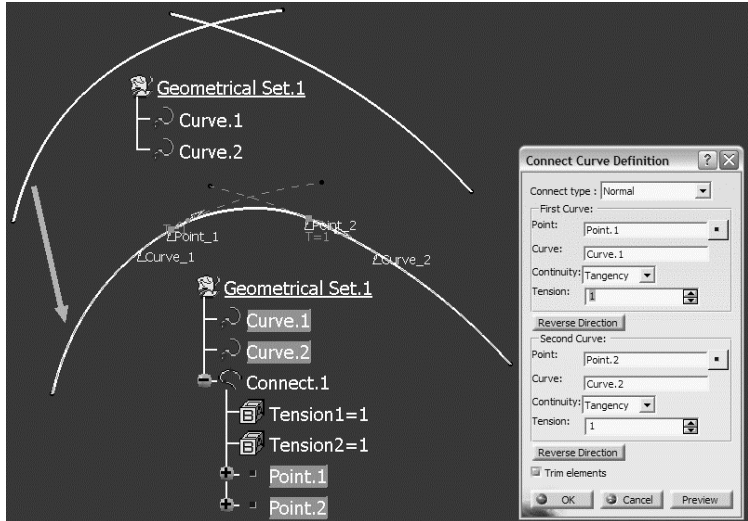
Konstrukcja krzywej sklejonej *Join.1*



Możliwa jest także taka sytuacja, w której dwie krzywe trzeba przyciąć i zdefiniować krzywą łączącą. Najprostszym rozwiązaniem jest w takim przypadku zastosowanie polecenia *Connect Curve*, które poza definicją punktów przycięcia obu krzywych umożliwia kontrolę rodzaju ciągłości krzywych pierwotnych (*Curve.1* i *Curve.2*) z krzywą łączącą *Connect.1* (rysunek 2.300).

**Rysunek 2.300.**

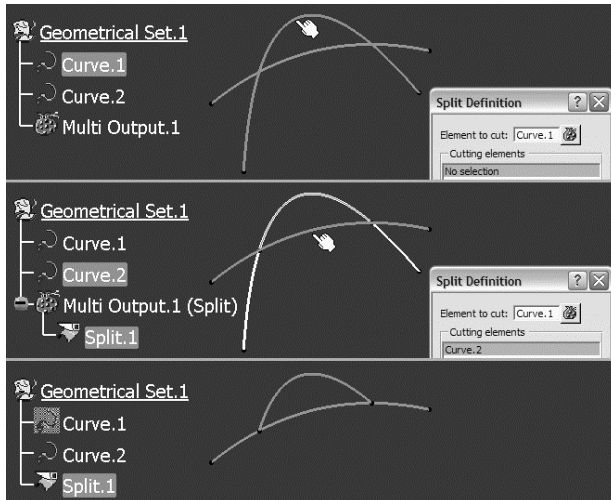
*Przycięcie i uzupełnienie odrzuconych fragmentów krzywych za pomocą polecenia *Connect Curve**



Chciałbym zwrócić uwagę Czytelnika na jeszcze jeden istotny, moim zdaniem, aspekt zastosowania polecenia *Split*. Wskazanie krzywej do odcięcia (pole *Element to cut* okna *Split Definition*) jest jednocześnie wskazaniem tej części krzywej, którą system powinien pozostawić po zakończeniu operacji *Split*. Na rysunku 2.301 wskazany (środkowy) fragment krzywej *Curve.1* jest rezultatem jej przycięcia za pomocą krzywej *Curve.2*.

**Rysunek 2.301.**

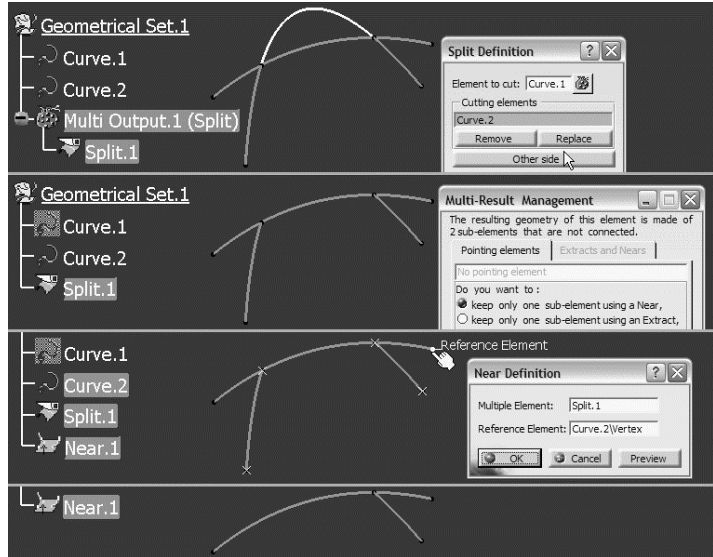
*Wskazanie krzywej do odcięcia jest jednocześnie wskazaniem, która część krzywej ma być zachowana po zatwierdzeniu polecenia *Split**



Gdyby rezultatem zastosowania polecenia *Split* miała być część krzywej *Curve.1* leżąca po przeciwnej stronie elementu tnącego (*Curve.2*), to wystarczy wcisnąć klawisz *Other side* (rysunek 2.302). Może się jednak zdarzyć, że krzywa wynikowa *Split.1* będzie złożona z kilku (na przykład dwóch) nieciągłych segmentów. W terminologii systemu CATIA V5 taki przypadek nosi nazwę *Non connex result*. W takim przypadku po zatwierdzeniu definicji krzywej *Split.1* system uruchomi automatycznie, omawiany już wcześniej, tryb *Multi-Result Management*, którego rezultatem będzie krzywa *Near.1* lub *Extract.1*.

### Rysunek 2.302.

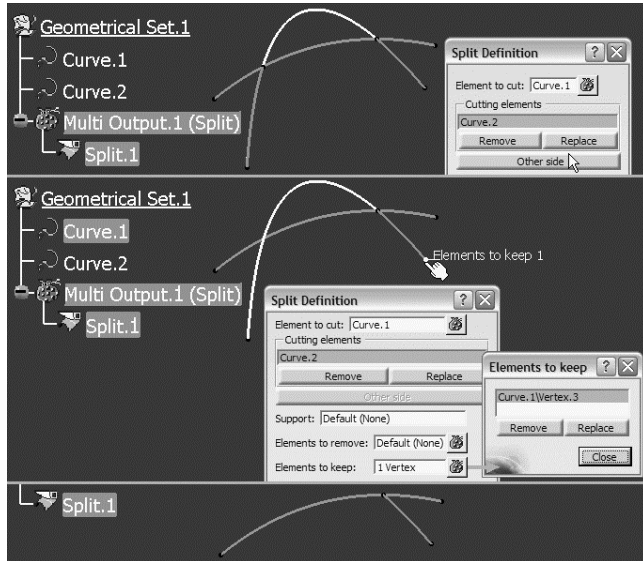
Przykład krzywej typu *Split* złożonej z dwóch nieciągłych fragmentów i reakcji systemu na taką niejednoznaczność



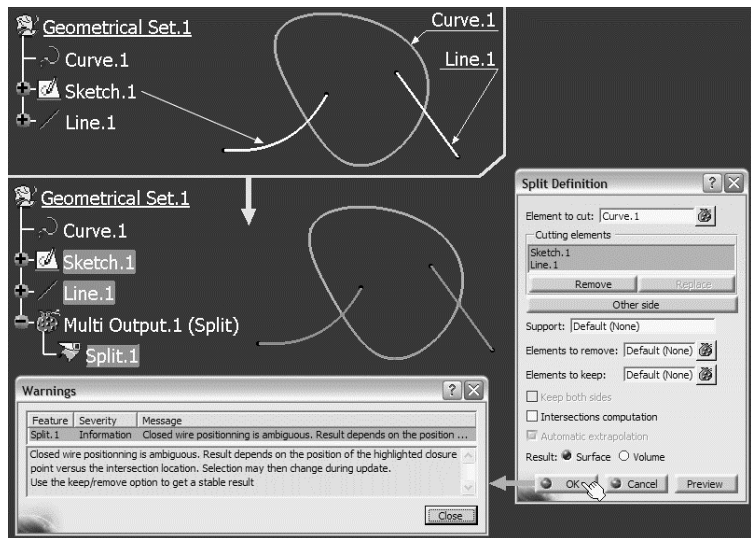
Aby uniknąć niejednoznaczności definicji krzywej *Split.1* i stosowania trybu *Multi-Result Management*, można w oknie *Split Definition* ustalić listę elementów do usunięcia (*Elements to remove*) lub listę elementów, które mają być zachowane (*Elements to keep*). Na przykład jeśli wskazany zostanie jeden z punktów granicznych krzywej *Curve.1* (*Curve.1|Vertex.3*) jako *Element to keep*, to wynikowa krzywa *Split.1* będzie zawierać tylko najbliższy fragment krzywej *Curve.1* po wybranej stronie elementu tnącego *Curve.2* (rysunek 2.303).

Polecenie *Split* jest zazwyczaj stosowane do przycinania krzywych otwartych. Gdyby trzeba było przyciąć krzywą zamkniętą, to potrzebny jest jeden element tnący, który ma dwa punkty przecięcia z krzywą przycinaną lub dwa elementy tnące. Na przykład (rysunek 2.304) krzywa *Curve.1* może być przycięta przez linię *Line.1* i krzywą *Sketch.1*. Niestety, w przypadku krzywej zamkniętej wskazanie części krzywej, która ma pozostać po odcięciu, czyli wskazanie punktu na krzywej, nie zawsze jest jednoznaczne, bo położenie tego punktu jest zależne od punktu początkowego tej krzywej. Tak, każda krzywa (także krzywa zamknięta) ma swój teoretyczny początek (*Closing Point*), czyli punkt, w którym współrzędna *U* krzywej ma wartość 0. Położenie tego punktu może się zmienić po wykonaniu zmian geometrycznych i dlatego system sugeruje w oknie *Warnings* zastosowanie omówionego wcześniej trybu *Keep/Remove*.

**Rysunek 2.303.**  
Zastosowanie trybu  
*Elements to keep* w  
definicji polecenia *Split*

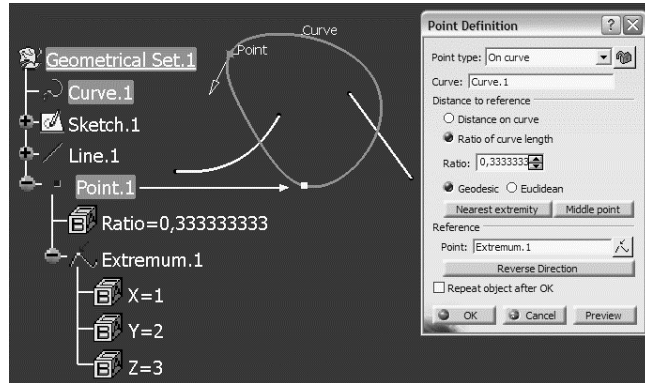


**Rysunek 2.304.**  
Próba odcięcia  
fragmentu krzywej  
zamkniętej



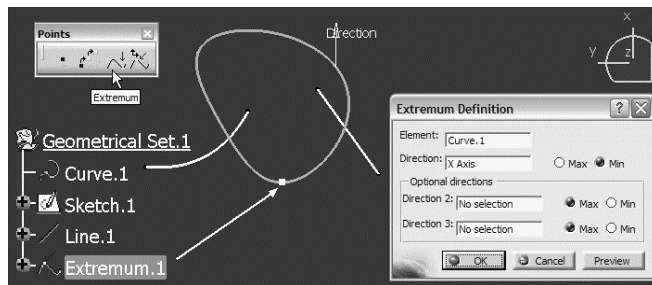
Ale w jaki sposób uniezależnić wybór części krzywej do odcięcia od potencjalnych, czyli nie do końca znanych, zmian konstrukcyjnych? Czy punkt *Point.1* (rysunek 2.305), zdefiniowany w  $\frac{1}{3}$  długości krzywej *Curve.1* od jej początku, może być zastosowany jako jednoznaczny wskaźnik tej części krzywej, którą trzeba pozostawić? Moim zdaniem nie, bo jego współrzędne są zależne od punktu *Extremum.1*, a ten z kolei jest punktem ekstremalnym w kierunku wyznaczonym przez arbitralnie ustalony wektor [1,2,3]. Takie współrzędne wektora kierunku są stosowane zawsze do wyznaczenia teoretycznego punktu początkowego krzywej zamkniętej. Skoro wektor kierunku nie ma nic wspólnego z krzywą, to wyznaczony na jego podstawie punkt *Extremum.1*, a także *Point.1* nie są w żaden sposób związane z otoczeniem geometrycznym krzywej *Curve.1*.

**Rysunek 2.305.**  
Konstrukcja punktu  
*Point.1* na krzywej  
*Curve.1*



Czy punkt *Point.1* musi być wyznaczony na podstawie „sztucznego” punktu początkowego krzywej *Curve.1*? Nie. Można przecież zdefiniować punkt przecięcia krzywej *Curve.1* z linią *Line.1*, a potem wskazać ten punkt przecięcia jako *Reference Point* w oknie *Point Definition*. Taka definicja punktu *Point.1* gwarantuje, że „podaży” on za swoimi „rodzicami”, czyli dostosuje się on do każdej zmiany kształtu krzywej *Curve.1* oraz linii *Line.1*. Procedurę konstrukcyjną punktu można uprościć, bo zamiast dwóch punktów (*Extremum.1* i *Point.1*) można zdefiniować tylko jeden. Na przykład jeśli krzywa *Curve.1* leży na płaszczyźnie *XY*, to wystarczy wyznaczyć punkt, którego współrzędna *X* jest minimalna (*Extremum.1* na rysunku 2.306).

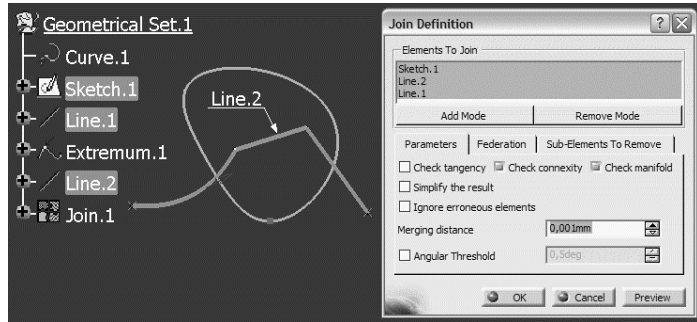
**Rysunek 2.306.**  
Definicja punktu  
*Extremum.1*,  
czyli punktu *Minimum*  
krzywej *Curve.1*  
w kierunku *X*



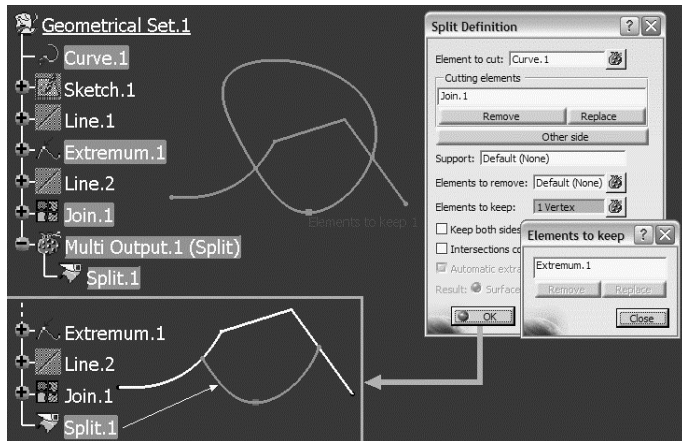
Pora powrócić do tematu głównego, czyli polecenia *Split*. W polu *Cutting elements* okna *Split Definition* można wskazać więcej niż jeden element tnący (na przykład *Sketch.1* i *Line.1*), ale lepszym rozwiązaniem jest „sklejenie” tych elementów w jeden obiekt. Jeśli wynik „sklejenia” (krzywa typu *Join*) ma być ciągły, to przed zastosowaniem polecenia *Join* należy wykreślić linię *Line.2* łączącą krzywe *Sketch.1* i *Line.1*. Krzywa *Join.1* może być wtedy zdefiniowana z aktywną opcją *Check continuity* (rysunek 2.307). Bez linii pomocniczej *Line.2* definicja krzywej *Join.1* jest możliwa, ale oczywiście z wyłączoną opcją *Check continuity*.

Po przygotowaniu wszystkich elementów pomocniczych (*Extremum.1* i *Join.1*) można powtórzyć próbę odcięcia krzywej zamkniętej *Curve.1* przez krzywą *Join.1* (rysunek 2.308). Aby uniknąć ostrzeżenia o niejednoznaczności w polu *Elements to keep* okna *Split Definition*, należy wskazać punkt *Extremum.1*.

**Rysunek 2.307.**  
Definicja krzywej sklejanej Join.1

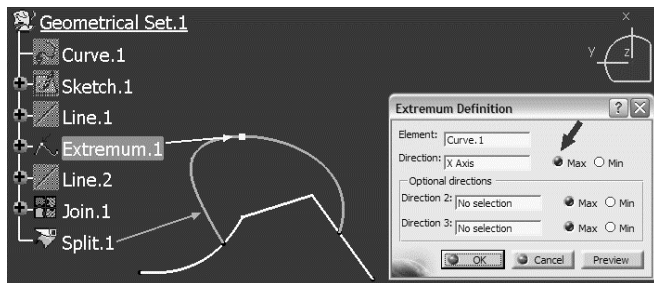


**Rysunek 2.308.**  
Przykład odcięcia części krzywej zamkniętej



Poprawność takiej konstrukcji łatwo sprawdzić, na przykład przez modyfikację punktu *Extremum.1*. Zamiana typu punktu z *Min* na *Max* i uruchomienie procedury *Update* spowoduje oczywiście zmianę tej części krzywej *Curve.1*, którą trzeba pozostawić (rysunek 2.309), bo zmieniło się położenie punktu *Extremum.1*.

**Rysunek 2.309.**  
Położenie punktu *Extremum.1* wpływa na rezultat przycięcia krzywej *Curve.1*



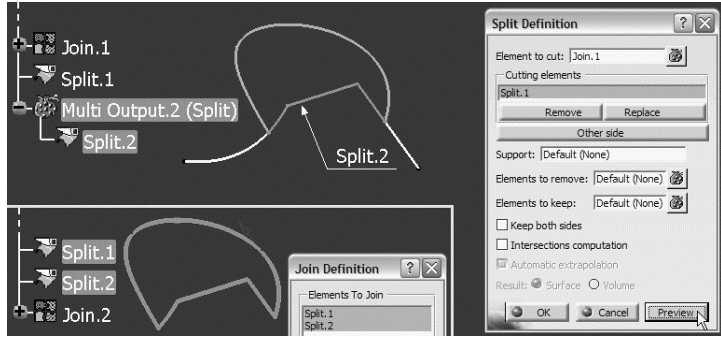
## Polecenie Trim

Omówione w poprzednim punkcie polecenie *Split* jest zazwyczaj początkiem pewnej standardowej procedury konstrukcyjnej, której celem jest definicja krzywej sklejonej z kilku wzajemnie ciągłych krzywych cząstkowych. Krzywe cząstkowe powstają w rezultacie

zastosowania różnych poleceń środowiska *GSD* i jeśli trzeba, są przycinane (polecenie *Split*) i wreszcie łączone w jedną krzywą zespoloną (polecenie *Join*). Na przykład krzywa *Split.1* z rysunku 2.309 może być zastosowana jako element tnący krzywą *Join.1*, aby w rezultacie można było zdefiniować krzywą zamkniętą *Join.2* zbudowaną z krzywych *Split.1* i *Split.2* (rysunek 2.310).

### Rysunek 2.310.

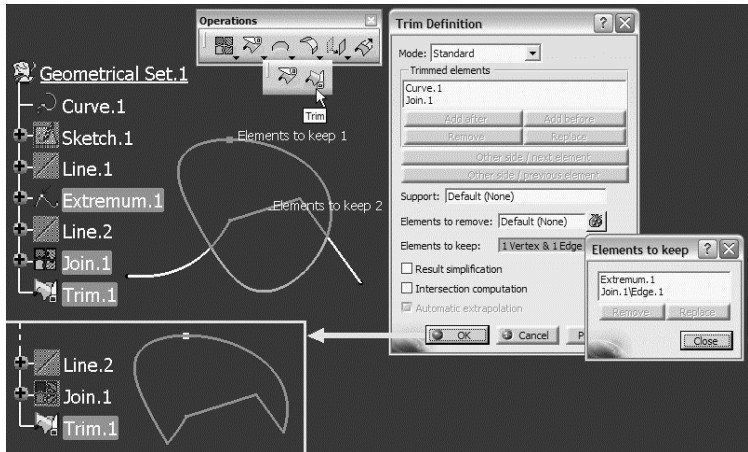
Przykład budowy krzywej zamkniętej *Join.2* zgodnie z procedurą *Split+Split+Join*



Polecenie *Trim* w trybie *Standard* upraszcza tę procedurę, bo umożliwia jednoczesne przycięcie i połączenie krzywych częściowych w jeden obiekt. Rezultat zastosowania polecenia *Trim* do krzywych *Curve.1* i *Join.1*, czyli krzywa *Trim.1* (rysunek 2.311), jest geometrycznie identyczny z krzywą *Join.2* z rysunku 2.310.

### Rysunek 2.311.

Przykład budowy krzywej zamkniętej *Trim.1* za pomocą polecenia *Trim*

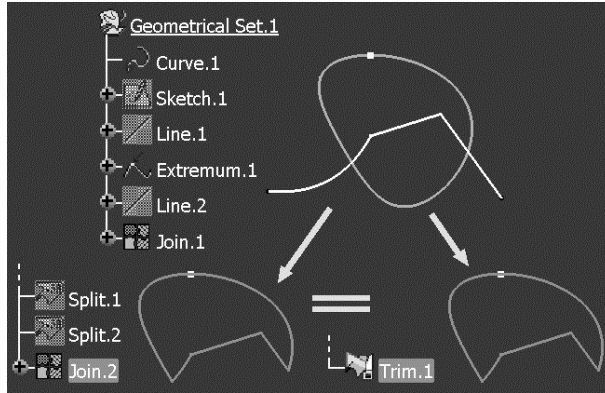


Dla tych samych elementów początkowych, czyli krzywych *Curve.1* i *Join.1*, można więc zdefiniować „umowne” równanie:  $Split.1 + Split.2 + Join.2 = Trim.1$  (rysunek 2.312).

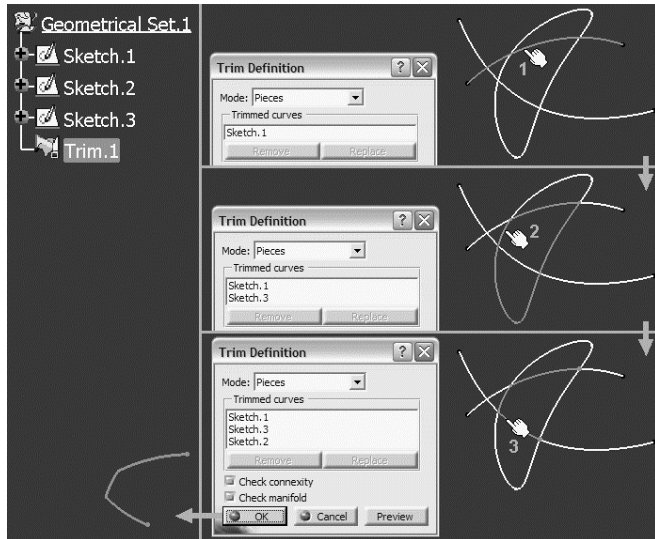
Polecenie *Trim* zastosowane w trybie *Pieces* buduje krzywą sklejoną z segmentów wskazanych krzywych wejściowych. Każde wskazanie krzywej oznacza wskazanie jej segmentu pomiędzy najbliższymi punktami przecięcia z innymi krzywymi (rysunek 2.313). Zastosowanie takiej metody pozwala pominąć czasochłonne przycinanie wielu krzywych w celu uzyskania jednej krzywej ciągłej.

**Rysunek 2.312.**

Zbudowane za pomocą różnych procedur konstrukcyjnych krzywe *Join.2* i *Trim.1* są geometrycznie identyczne

**Rysunek 2.313.**

Zastosowanie polecenia *Trim* w trybie *Pieces*

**Wygladzanie krzywych**

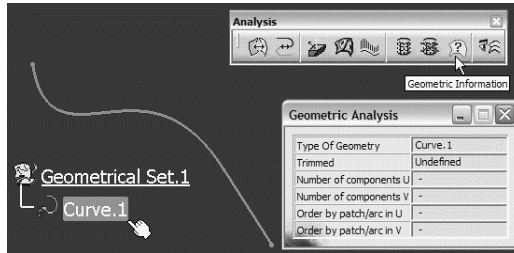
Krzywe elementarne, których definicja jest realizowana przez polecenia *Spline*, *Connect Curve* czy *Conic*, są zawsze krzywymi ciągłymi zgodnie z kryterium G2 (ciągłość krzywizny). Krzywe konstrukcyjne, które powstają w wyniku przycięcia i sklejenia krzywych cząstkowych, a także jako krzywe przecięcia powierzchni lub rzutowania krzywej na powierzchnię, nie zawsze spełniają wysokie wymagania w zakresie ciągłości. Także krzywe importowane z innych systemów CAD, czy to bezpośrednio, czy też poprzez formaty neutralne (*IGES*, *STEP*), są obciążone pewnymi błędami konwersji, które wpływają na jakość krzywej. Dlatego obok sztuki definiowania krzywych konstruktor musi posiadać umiejętność analizy jakości krzywych i metod wygładzania.

Jeśli nie jest znany sposób definicji krzywej, bo jest to krzywa importowana lub zdefiniowana jako *Datum*, to pierwszym zadaniem konstruktora powinna być analiza geometryczna. Taka analiza może być przeprowadzona wizualnie i dodatkowo może być wspomagana



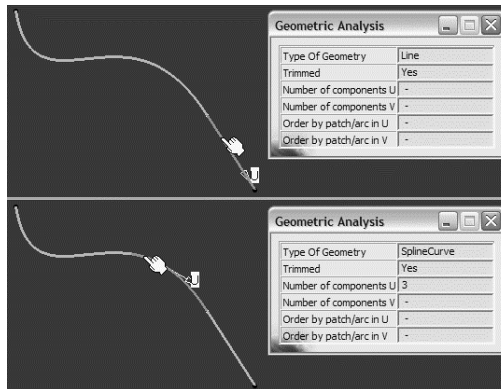
poleceniem *Geometric Information* (rysunek 2.314). Niestety, w przypadku krzywych sklejanych informacje dostępne w oknie *Geometric Analysis* po wskazaniu całej krzywej (*Curve.1* w drzewie strukturalnym modelu) są raczej skąpe, żeby nie powiedzieć żadne.

**Rysunek 2.314.**  
*Analiza geometryczna krzywej sklejanej*



Znacznie więcej informacji uzyskamy po wskazaniu jednego z segmentów krzywej sklejanej (rysunek 2.315), bo wtedy wiadomo, że krzywa *Curve.1* składa się z odcinka linii prostej (*Type Of Geometry = Line*) i krzywej typu *Spline* (*Type Of Geometry = SplineCurve*) złożonej z trzech komponentów (*Number of components U = 3*). W sumie krzywa *Curve.1* składa się z 4 komponentów.

**Rysunek 2.315.**  
*Analiza geometryczna segmentów krzywej sklejanej*



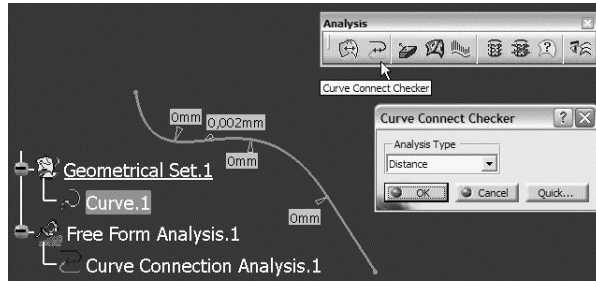
Znacznie więcej informacji o krzywej, zwłaszcza w kontekście analizy jej ciągłości, dostarcza polecenie *Curve Connect Checker* (rysunek 2.316). Jeśli dla tej samej krzywej *Curve.1* wykonamy tę analizę w trybie *Analysis Type = Distance*, to uzyskamy obraz ciągłości geometrycznej (G0). Jeśli krzywa *Curve.1* została sklejona z 4 komponentów, to zakładając, że każdy z nich jest ciągły, powinniśmy uzyskać analizę ciągłości w 3 punktach wspólnych. Tu jednak jest inaczej, bo system wskazuje 4 analizowane punkty, z których jeden ma nieciągłość geometryczną o wielkości  $0,002\text{ mm}$ .

Próbując zidentyfikować problem, można wyodrębnić krzywą typu *Spline*, która obok odcinka linii prostej jest jednym z komponentów krzywej *Curve.1*. Takie zadanie może być wykonane za pomocą polecenia *Extract* po wskazaniu segmentu liniowego i włączeniu opcji *Complementary mode* (rysunek 2.317).

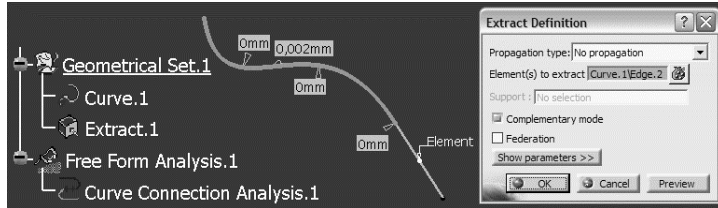
Wizualna analiza obszaru, w którym system wskazuje na niewielką, bo bliską tolerancji nieciągłość geometryczną, nic nie daje. Dlaczego? Tu przypomnę jedynie topologiczną

**Rysunek 2.316.**

Analiza ciągłości geometrycznej krzywej *Curve.1*

**Rysunek 2.317.**

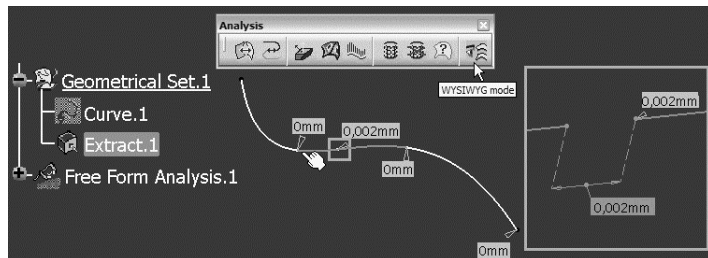
Wyodrębnienie krzywej typu *Spline* ze struktury krzywej *Curve.1*



ciągłość krzywych sklejanych (patrz polecenie *Join*) kontrolowaną przez wartość parametru *Merging distance*. Dopiero włączenie kolejnego narzędzia (rysunek 2.318), jakim jest precyzyjny sposób wizualizacji geometrii *WYSIWYG mode* (*What You See Is What You Get*), pozwala w odpowiednim powiększeniu zobaczyć i zmierzyć wykrytą przez *Curve Connect Checker* nieciągłość krzywej *Curve.1*.

**Rysunek 2.318.**

Wizualizacja nieciągłości krzywej w trybie *WYSIWYG*

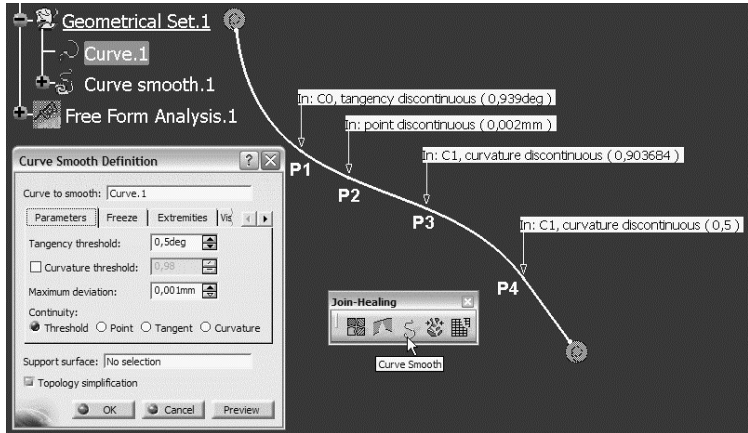


Jak naprawić taką nieciągłość? Jednym z rozwiązań może być rekonstrukcja środkowego segmentu krzywej *Extract.1* za pomocą polecenia *Spline* lub *Connect Curve*, z zachowaniem wymaganej ciągłości do pozostałych segmentów. Taka metoda oznacza oczywiście dekompozycję krzywej na segmenty elementarne, rekonstrukcję tych, które są nieciągłe, oraz powtórne sklejenie segmentów w jedną ciągłą krzywą. Mówiąc wprost, wymaga to dużo benedyktyńskiej pracy, która może być wykonana automatycznie za pomocą polecenia *Curve Smooth* (rysunek 2.319). Zanim zostanie wykonane wygładzenie wskazanej krzywej, system określa warunki początkowe:

- ◆ W punkcie P1 — krzywa *Curve.1* ma ciągłość G0 i nieciągłość stycznej ( $0,939\text{ deg}$ ),
- ◆ W punkcie P2 — krzywa *Curve.1* ma nieciągłość geometryczną ( $0,002\text{ mm}$ ),
- ◆ W punkcie P3 — krzywa *Curve.1* ma ciągłość G1 i nieciągłość krzywizny ( $0,903684$ ),
- ◆ W punkcie P4 — krzywa *Curve.1* ma ciągłość G1 i nieciągłość krzywizny ( $0,5$ ).

**Rysunek 2.319.**

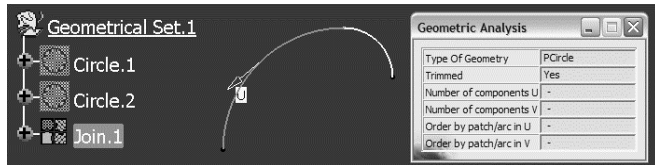
Wstępna analiza ciągłości krzywej po uruchomieniu polecenia *Curve Smooth*



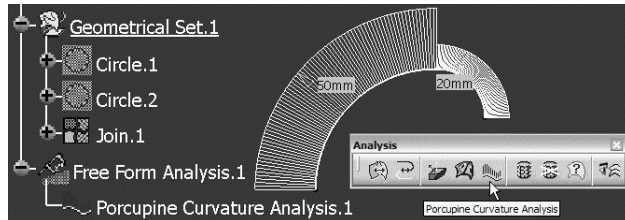
O ile nie ma problemu z interpretacją nieciągłości geometrycznej ( $0,002\text{ mm}$ ) lub nieciągłość styczności ( $0,939\text{ deg}$ ), to nieciągłość krzywizny wyrażona liczbowo ( $0,903684$  i  $0,5$ ) powinna być wyjaśniona. Rozważmy przykład krzywej *Join.1* (rysunek 2.320) sklejonej z dwóch łuków okręgu o różnych promieniach (rysunek 2.321).

**Rysunek 2.320.**

Krzywa *Join.1* sklejona z dwóch łuków okręgu

**Rysunek 2.321.**

Analiza rozkładu krzywizny krzywej *Join.1*

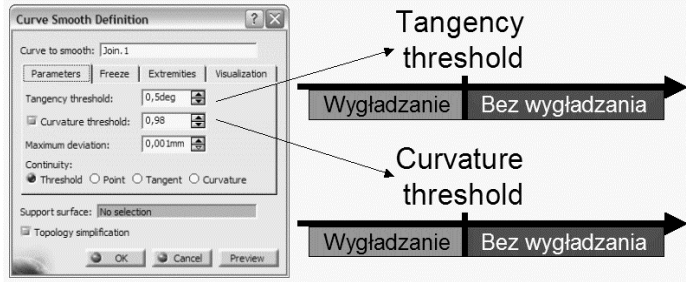


Nieciągłość krzywizny jest obliczana według wzoru  $Nk = |R1 - R2| / R2$ , gdzie  $R1$  i  $R2$  są promieniami krzywizny obu krzywych w ich punkcie wspólnym. Dla przykładu z rysunku 2.321 mamy  $Nk = |20 - 50| / 50 = 0,6$ . Liczbowo wyrażona nieciągłość to jeszcze za mało, bo konieczne jest także ustalenie wartości granicznej, powyżej której wygładzanie krzywej nie powinno być wykonywane (rysunek 2.322). Zupełnie podobnie jak w przypadku polecenia *Join* i parametru *Merging distance*. Gdyby nieciągłość krzywizny była niewielka, na przykład  $Nk = 0,99$ , i tolerancję ciągłości krzywizny pozostawiono by na poziomie standardowym (*Curvature threshold* =  $0,98$ ), to polecenie *Curve Smooth* nie zmieni rozkładu krzywizny wygładzanej krzywej.

Stosownie do wartości parametrów *Tangency threshold*, *Curvature threshold* i aktualnej nieciągłości krzywej algorytm polecenia *Curve Smooth* modyfikuje kształt krzywej wygładzanej. Maksymalna dopuszczalna deformacja krzywej jest określona wartością parametru

**Rysunek 2.322.**

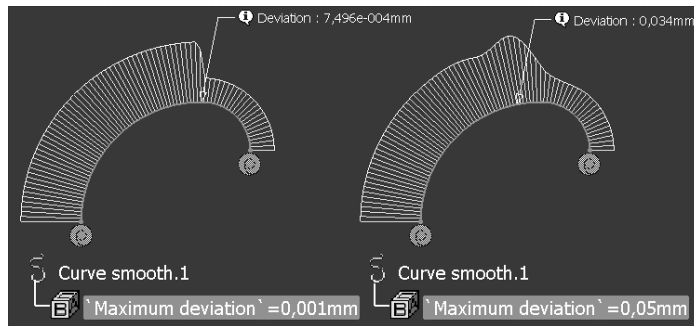
Tolerancje ciągłości stycznych i krzywizny stosowane przez algorytm wygładzania krzywej polecenia *Curve Smooth*



*Maximum deviation.* Im większa wartość tego parametru, tym więcej swobody ma system w poszukiwaniu najlepszej krzywej i tym większe jest prawdopodobieństwo jednorodnego (bez gwałtownych zmian) rozkładu zmian krzywizny krzywej (rysunek 2.323).

**Rysunek 2.323.**

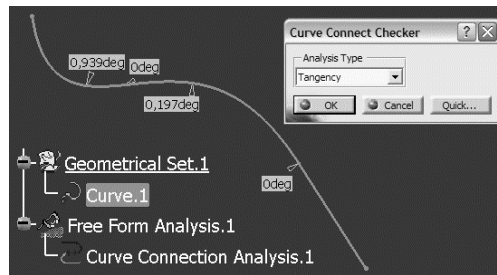
Porównanie rozkładów zmian krzywizny dla różnych wartości parametru *Maximum deviation*



Po wyjaśnieniu terminów i zasad wygładzania krzywej powróćmy do analizy ciągłości i wygładzania krzywej sklejanej *Curve.1* z rysunku 2.319. Polecenie *Curve Connect Checker* pozwala analizować nie tylko ciągłość geometryczną krzywej, ale także ciągłość stycznych (rysunek 2.324) i ciągłość krzywizny.

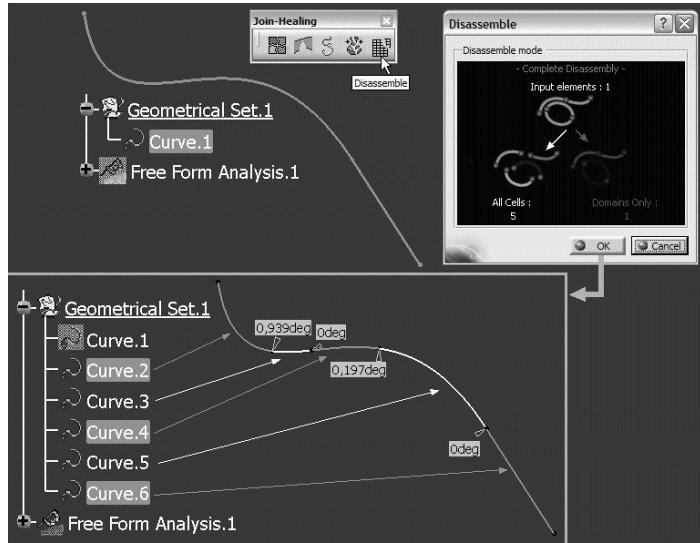
**Rysunek 2.324.**

Analiza ciągłości stycznych krzywej *Curve.1*



Algorytm wygładzania polecenia *Curve Smooth* modyfikuje krzywą nie tylko w taki sposób, by zapewnić jej ciągłość topologiczną, ale przede wszystkim geometryczną ciągłość jej elementów składowych. Dlatego czasami warto wykonać dekompozycję krzywej, aby zobaczyć, jak algorytm wygładzania modyfikuje kształt krzywych cząstkowych. Polecenie *Disassemble* (rysunek 2.325) zastosowane do krzywej *Curve.1* tworzy 5 krzywych cząstkowych (*Curve.2* – *Curve.6*), z których każda jest „wewnętrznie” ciągła według kryterium G2. Analiza ciągłości powinna być więc wykonana jedynie w punktach wspólnych tych krzywych lub inaczej w węzłach krzywej sklejanej.

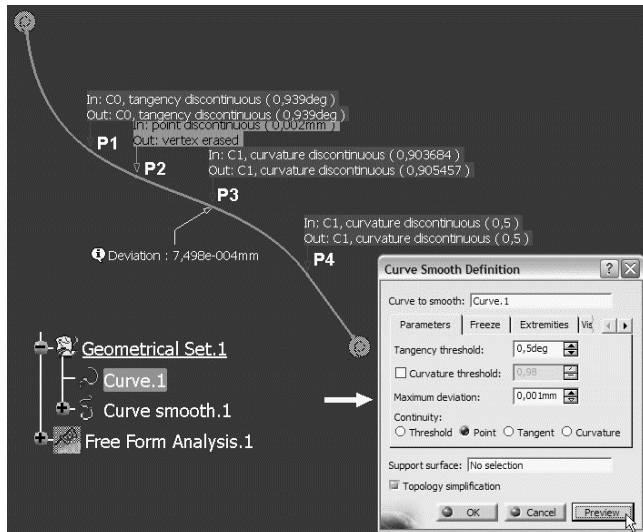
**Rysunek 2.325.**  
Zastosowanie polecenia *Disassemble* do dekompozycji krzywej *Curve.1*



Po ustaleniu wartości tolerancji granicznych (*Tangency threshold* i *Curvature threshold*) oraz maksymalnej dopuszczalnej deformacji krzywej (*Maximum deviation*) można rozpocząć proces wygładzania krzywej *Curve.1*, którego celem może być osiągnięcie jednego z trzech rodzajów ciągłości:

1. *Continuity = Point* — ciągłość geometryczna krzywej (rysunek 2.326).

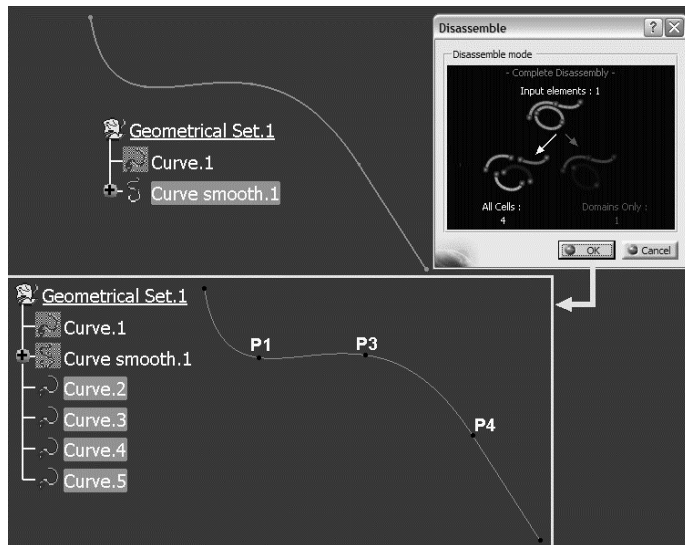
**Rysunek 2.326.**  
Wygładzenie krzywej *Curve.1* w trybie *Continuity = Point*



System proponuje krzywą ciągłą co najmniej według kryterium G0, której maksymalna odchyłka od krzywej pierwotnej (*Deviation: 7,498e-004 mm*) jest mniejsza od *Maximum deviation = 0,001 mm*. Wstępne rezultaty wygładzania krzywej *Curve.1* (klawisz *Preview* w oknie *Curve Smooth Definition*) można interpretować następująco:

- a. W punkcie P1 stan przed (*In: C0, tangency discontinuous (0,939 deg)*) i po wygładzaniu (*Out: C0, tangency discontinuous (0,939 deg)*) nie zmienił się. Nie ma zmiany na lepsze (problem nieciągłości nie został naprawiony) i dlatego opis jest umieszczony na czerwonym tle. Krzywa *Curve smooth.1* ma w tym punkcie ciągłość G0 oraz nieciągłość stycznych o wielkości 0,939 deg.
- b. W punkcie P2 stan przed (*In: point discontinuous (0,002 mm)*) i po wygładzaniu (*Out: vertex erased*) zmienił się. Nieciągłość geometryczna krzywej została naprawiona i dlatego opis jest umieszczony na zielonym tle. Węzeł P2 krzywej *Curve.1* został usunięty (*vertex erased*), a to oznacza, że krzywe *Curve.3* i *Curve.4* z rysunku 2.325 zostały zastąpione jedną krzywą ciągłą. Taka ingerencja w topologiczną strukturę krzywej *Curve.1* jest możliwa wtedy, gdy aktywny jest tryb *Topology simplification*. Potwierdzenie takiej modyfikacji (redukcji liczby krzywych cząstkowych) łatwo zobaczyć po zaakceptowaniu definicji krzywej *Curve smooth.1* i zastosowaniu polecenia *Disassemble* (rysunek 2.327).

**Rysunek 2.327.**  
 Dekompozycja krzywej  
*Curve smooth.1*

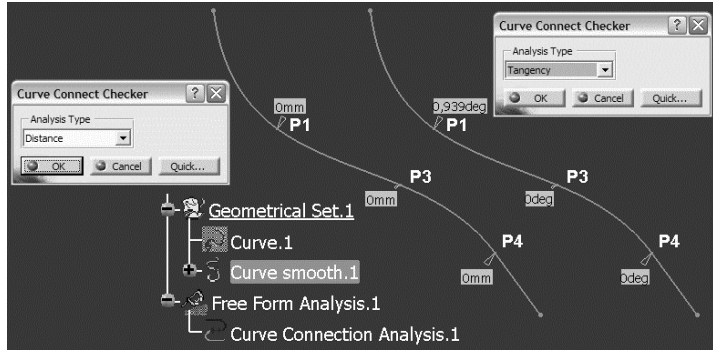


- c. W punkcie P3 stan przed (*In: C1, curvature discontinuous*) i po wygładzaniu (*Out: C1, curvature discontinuous*) w zasadzie nie zmienił się, chociaż nieciągłość krzywizny zmieniła się nieznacznie z 0,903684 na 0,905457. Problem nieciągłości nie został naprawiony i dlatego opis jest umieszczony na czerwonym tle. Krzywa *Curve smooth.1* ma w tym punkcie ciągłość G1.
- d. W punkcie P4 stan przed (*In: C1, curvature discontinuous (0,5)*) i po wygładzaniu (*Out: C1, curvature discontinuous (0,5)*) nie zmienił się. Problem nieciągłości nie został naprawiony i dlatego opis jest umieszczony na czerwonym tle. Krzywa *Curve smooth.1* ma w tym punkcie ciągłość G1.

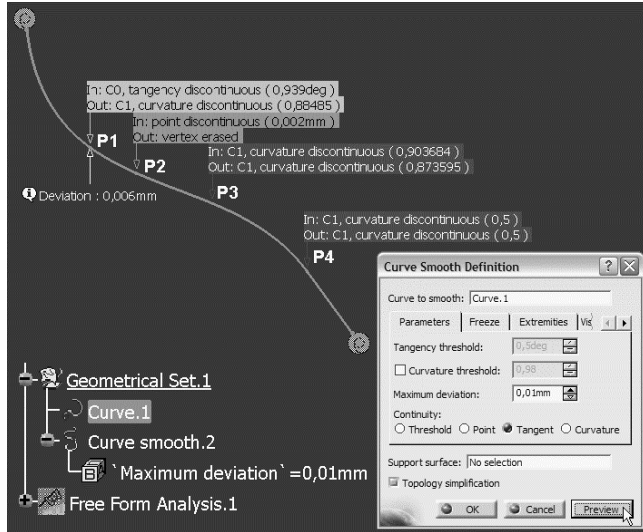
Krzywa *Curve smooth.1* jest krzywą ciągłą według kryterium G0, ale nie G1 (rysunek 2.328).

2. *Continuity = Tangent* — ciągłość stycznych krzywej (rysunek 2.329).

**Rysunek 2.328.**  
Wyniki analizy ciągłości  
krzywej *Curve smooth.1*



**Rysunek 2.329.**  
Wyglądzenie krzywej  
*Curve.2* w trybie  
*Continuity = Tangent*



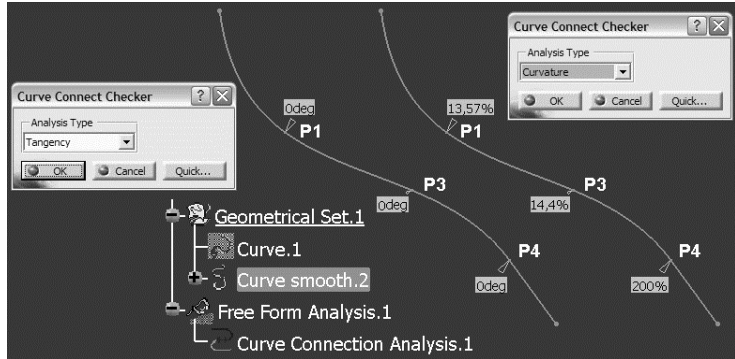
System proponuje krzywą ciągłą według kryterium G1, której maksymalna odchyłka od krzywej pierwotnej (*Deviation: 0,006 mm*) jest mniejsza od *Maximum deviation = 0,01 mm*. Wstępne rezultaty wygładzania krzywej *Curve.1* można interpretować następująco:

- a. W punkcie P1 stan przed (*In: C0, tangency discontinuous (0,939 deg)*) i po wygładzaniu (*Out: C1, curvature discontinuous (0,88485)*) zmienił się. Problem nieciągłości został częściowo naprawiony (zmiana z G0 na G1) i dlatego opis jest umieszczony na żółtym tle. Krzywa *Curve smooth.2* ma w tym punkcie ciągłość G1 oraz nieciągłość krzywizny o wielkości 0,88485.
- b. W punkcie P2 komentarz jest taki sam jak dla *Continuity = Point*.
- c. W punkcie P3 stan przed (*In: C1, curvature discontinuous*) i po wygładzaniu (*Out: C1, curvature discontinuous*) w zasadzie nie zmienił się, chociaż nieciągłość krzywizny zmieniła się nieznacznie z 0,903684 na 0,873595. Problem nieciągłości nie został naprawiony i dlatego opis jest umieszczony na czerwonym tle. Krzywa *Curve smooth.2* ma w tym punkcie ciągłość G1.
- d. W punkcie P4 komentarz jest taki sam jak dla *Continuity = Point*.

Krzywa *Curve smooth.2* jest krzywą ciągłą według kryterium G1, ale nie G2 (rysunek 2.330).

**Rysunek 2.330.**

Wyniki analizy ciągłości krzywej *Curve smooth.2*

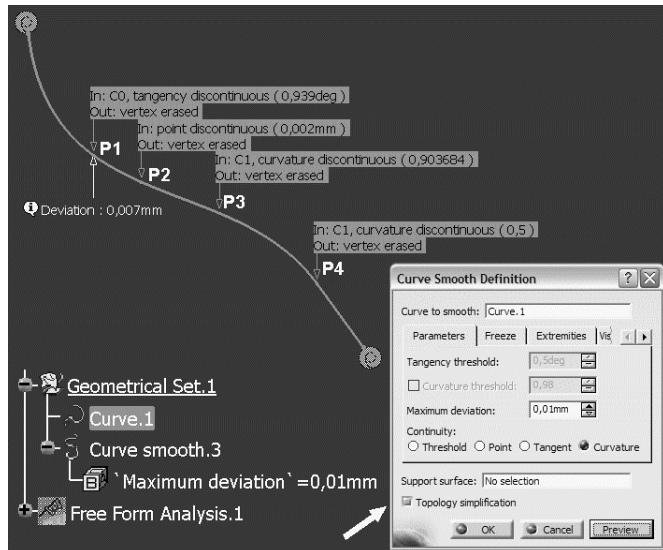


### 3. Continuity = Curvature — ciągłość krzywizny krzywej.

- a. Jeśli opcja *Topology simplification* jest aktywna (rysunek 2.331), to system proponuje krzywą ciągłą według kryterium G2, której maksymalna odchyłka od krzywej pierwotnej (*Deviation: 0,007 mm*) jest mniejsza od *Maximum deviation = 0,01 mm*. Wstępne rezultaty wygładzania krzywej *Curve.1* można interpretować następująco:

**Rysunek 2.331.**

Wygładzenie krzywej *Curve.3* w trybie *Continuity = Curvature* z aktywną opcją *Topology simplification*

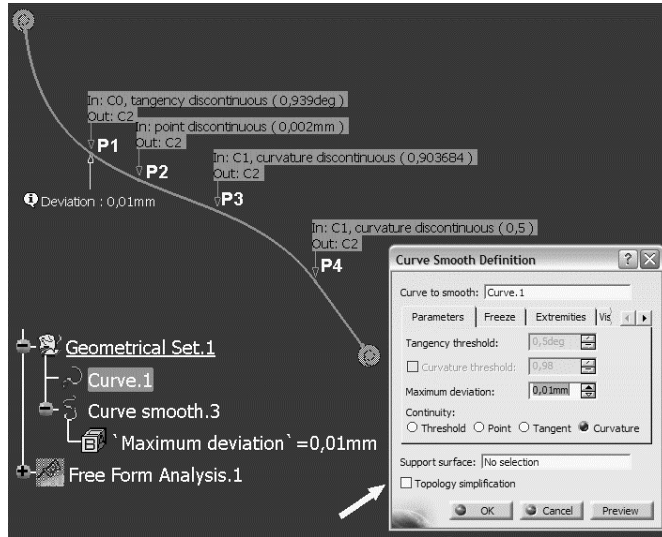


- ◆ Krzywa *Curve smooth.3* ma ciągłość G2 na całej swojej długości (opisy wszystkich punktów są umieszczone na zielonym tle).
  - ◆ Wszystkie węzły krzywej *Curve.1* zostały usunięte i dlatego krzywa *Curve smooth.3* jest krzywą typu *NUPBS* (rysunek 2.333).
- b. Jeśli opcja *Topology simplification* jest wyłączona (rysunek 2.332), to system proponuje krzywą ciągłą według kryterium G2, której maksymalna odchyłka

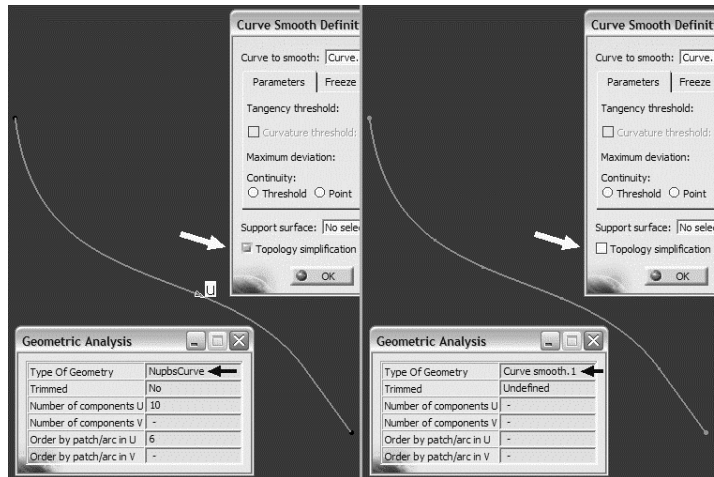


**Rysunek 2.332.**

Wyglądzenie krzywej  
Curve.3 w trybie  
Continuity = Curvature  
z wyłączoną opcją  
Topology simplification

**Rysunek 2.333.**

Wpływ aktywności opcji  
Topology simplification  
na model matematyczny  
krzywej Curve smooth.3



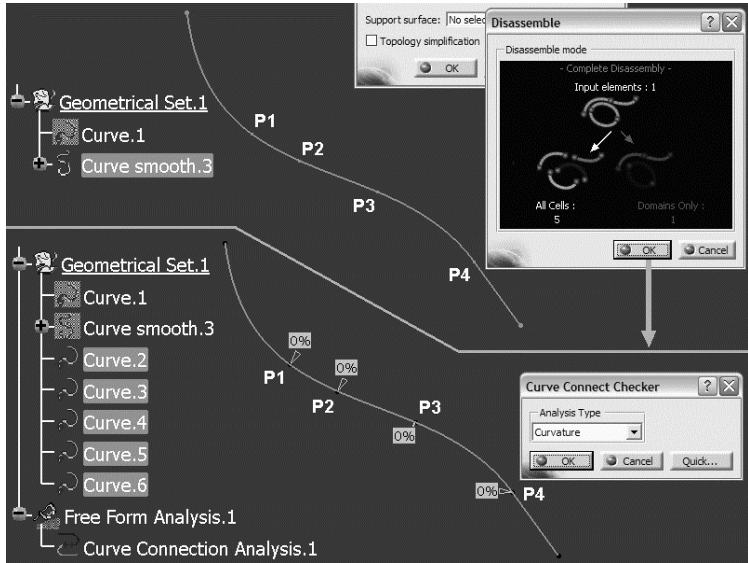
od krzywej pierwotnej (*Deviation: 0,01 mm*) jest równa *Maximum deviation = 0,01 mm*. Wstępne rezultaty wyglądzania krzywej *Curve.1* można interpretować następująco:

- ♦ Krzywa *Curve smooth.3* ma ciągłość G2 na całej swojej długości (opisy wszystkich punktów są umieszczone na zielonym tle).
- ♦ System zachował topologię krzywej pierwotnej (wszystkie węzły krzywej *Curve.1*) i dlatego krzywa *Curve smooth.3* jest „klasyczną” krzywą sklejaną (rysunek 2.333). Brak pełnej swobody modyfikacji kształtu krzywej w pobliżu punktów węzłowych jest powodem maksymalnej deformacji (*Deviation = 0,01 mm = Maximum deviation*).

Dekompozycja krzywej *Curve smooth.3* zdefiniowanej z wyłączoną opcją *Topology simplification* potwierdza zachowanie topologicznej struktury krzywej *Curve.1* oraz ciągłość krzywizny jej wszystkich krzywych cząstkowych (rysunek 2.334).

**Rysunek 2.334.**

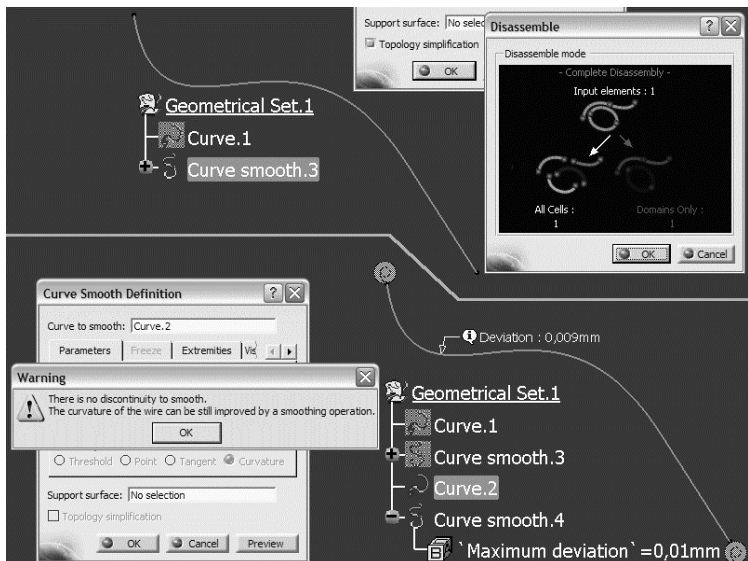
Dekompozycja krzywej *Curve smooth.3* i analiza ciągłości jej krzywych cząstkowych



Analogicznie dla krzywej *Curve smooth.3* zdefiniowanej z aktywną opcją *Topology simplification* jej dekompozycja oznacza stworzenie tylko jednej krzywej cząstkowej *Curve.2* (rysunek 2.335). Takiej pojedynczej krzywej nie ma powodu analizować pod kątem jej wewnętrznej ciągłości, ale można ją jeszcze wygładzić. Po zastosowaniu polecenia *Curve Smooth* do krzywej *Curve.2* w oknie *Warning* obok ostrzeżenia o braku nieciągłości jest też informacja o tym, że rozkład krzywizny może być jeszcze poprawiony.

**Rysunek 2.335.**

Dekompozycja krzywej *Curve smooth.3* i kolejna poprawa rozkładu jej krzywizny

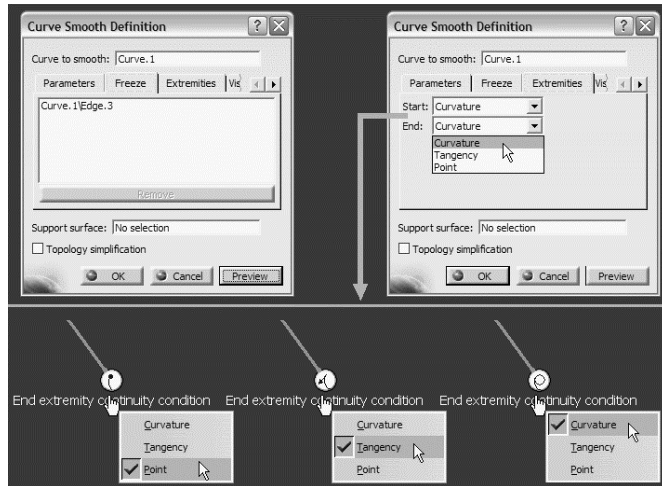




W każdym przypadku proces wygładzania krzywej powinien być prowadzony równoległe z analizą rozkładu krzywizny (patrz punkt „Polecenie Porcupine Curvature Analysis”).

Proces wygładzania krzywej powinien być zawsze ograniczony wymaganiami konstrukcyjnymi. Przykłady takich dodatkowych ograniczeń (rysunek 2.336) to „zamrożenie” wskazanego segmentu krzywej (zakładka *Freeze*) oraz ustalenie rodzaju ciągłości w punktach skrajnych wygładzanej krzywej (zakładka *Extremities* lub menu kontekstowe punktu skrajnego krzywej).

**Rysunek 2.336.**  
Definiowanie ograniczeń  
procesu wygładzania  
krzywej



W zakresie wstępnej wizualizacji wyników, szczególnie w przypadku krzywych o dużej liczbie punktów węzłowych, bardzo przydatny może być wybór sposobu wizualizacji (zakładka *Visualization* okna *Curve Smooth Definition* na rysunku 2.337):

- ♦ Interaktywny (*Display information interactively*), w którym pokazana jest informacja o rodzaju ciągłości dla wskazanego punktu węzłowego krzywej przed (*In:*) i po (*Out:*) wykonaniu polecenia.
- ♦ W kolejności punktów węzłowych (*Display information sequentially*), w którym opis dotyczy punktu następnego (*Next*) lub poprzedniego (*Previous*).