

# 14. 3D Object Representation




## 3D Object Representation

- Polyhedron
- Quadric Surfaces
- Blobby Object
- Splines
- Sweep Representation
- Constructive Solid Geometry

- CG 의 3대 분야
  - Modeling: 수학(기하) 기반 → *2D* *3D* *4D* *5D* *6D* *7D* *8D* *9D* *10D* *11D* *12D* *13D* *14D* *15D* *16D* *17D* *18D* *19D* *20D* *21D* *22D* *23D* *24D* *25D* *26D* *27D* *28D* *29D* *30D* *31D* *32D* *33D* *34D* *35D* *36D* *37D* *38D* *39D* *40D* *41D* *42D* *43D* *44D* *45D* *46D* *47D* *48D* *49D* *50D* *51D* *52D* *53D* *54D* *55D* *56D* *57D* *58D* *59D* *60D* *61D* *62D* *63D* *64D* *65D* *66D* *67D* *68D* *69D* *70D* *71D* *72D* *73D* *74D* *75D* *76D* *77D* *78D* *79D* *80D* *81D* *82D* *83D* *84D* *85D* *86D* *87D* *88D* *89D* *90D* *91D* *92D* *93D* *94D* *95D* *96D* *97D* *98D* *99D* *100D* *101D* *102D* *103D* *104D* *105D* *106D* *107D* *108D* *109D* *110D* *111D* *112D* *113D* *114D* *115D* *116D* *117D* *118D* *119D* *120D* *121D* *122D* *123D* *124D* *125D* *126D* *127D* *128D* *129D* *130D* *131D* *132D* *133D* *134D* *135D* *136D* *137D* *138D* *139D* *140D* *141D* *142D* *143D* *144D* *145D* *146D* *147D* *148D* *149D* *150D* *151D* *152D* *153D* *154D* *155D* *156D* *157D* *158D* *159D* *160D* *161D* *162D* *163D* *164D* *165D* *166D* *167D* *168D* *169D* *170D* *171D* *172D* *173D* *174D* *175D* *176D* *177D* *178D* *179D* *180D* *181D* *182D* *183D* *184D* *185D* *186D* *187D* *188D* *189D* *190D* *191D* *192D* *193D* *194D* *195D* *196D* *197D* *198D* *199D* *200D* *201D* *202D* *203D* *204D* *205D* *206D* *207D* *208D* *209D* *210D* *211D* *212D* *213D* *214D* *215D* *216D* *217D* *218D* *219D* *220D* *221D* *222D* *223D* *224D* *225D* *226D* *227D* *228D* *229D* *230D* *231D* *232D* *233D* *234D* *235D* *236D* *237D* *238D* *239D* *240D* *241D* *242D* *243D* *244D* *245D* *246D* *247D* *248D* *249D* *250D* *251D* *252D* *253D* *254D* *255D* *256D* *257D* *258D* *259D* *260D* *261D* *262D* *263D* *264D* *265D* *266D* *267D* *268D* *269D* *270D* *271D* *272D* *273D* *274D* *275D* *276D* *277D* *278D* *279D* *280D* *281D* *282D* *283D* *284D* *285D* *286D* *287D* *288D* *289D* *290D* *291D* *292D* *293D* *294D* *295D* *296D* *297D* *298D* *299D* *300D* *301D* *302D* *303D* *304D* *305D* *306D* *307D* *308D* *309D* *310D* *311D* *312D* *313D* *314D* *315D* *316D* *317D* *318D* *319D* *320D* *321D* *322D* *323D* *324D* *325D* *326D* *327D* *328D* *329D* *330D* *331D* *332D* *333D* *334D* *335D* *336D* *337D* *338D* *339D* *340D* *341D* *342D* *343D* *344D* *345D* *346D* *347D* *348D* *349D* *350D* *351D* *352D* *353D* *354D* *355D* *356D* *357D* *358D* *359D* *360D* *361D* *362D* *363D* *364D* *365D* *366D* *367D* *368D* *369D* *370D* *371D* *372D* *373D* *374D* *375D* *376D* *377D* *378D* *379D* *380D* *381D* *382D* *383D* *384D* *385D* *386D* *387D* *388D* *389D* *390D* *391D* *392D* *393D* *394D* *395D* *396D* *397D* *398D* *399D* *400D* *401D* *402D* *403D* *404D* *405D* *406D* *407D* *408D* *409D* *410D* *411D* *412D* *413D* *414D* *415D* *416D* *417D* *418D* *419D* *420D* *421D* *422D* *423D* *424D* *425D* *426D* *427D* *428D* *429D* *430D* *431D* *432D* *433D* *434D* *435D* *436D* *437D* *438D* *439D* *440D* *441D* *442D* *443D* *444D* *445D* *446D* *447D* *448D* *449D* *450D* *451D* *452D* *453D* *454D* *455D* *456D* *457D* *458D* *459D* *460D* *461D* *462D* *463D* *464D* *465D* *466D* *467D* *468D* *469D* *470D* *471D* *472D* *473D* *474D* *475D* *476D* *477D* *478D* *479D* *480D* *481D* *482D* *483D* *484D* *485D* *486D* *487D* *488D* *489D* *490D* *491D* *492D* *493D* *494D* *495D* *496D* *497D* *498D* *499D* *500D* *501D* *502D* *503D* *504D* *505D* *506D* *507D* *508D* *509D* *510D* *511D* *512D* *513D* *514D* *515D* *516D* *517D* *518D* *519D* *520D* *521D* *522D* *523D* *524D* *525D* *526D* *527D* *528D* *529D* *530D* *531D* *532D* *533D* *534D* *535D* *536D* *537D* *538D* *539D* *540D* *541D* *542D* *543D* *544D* *545D* *546D* *547D* *548D* *549D* *550D* *551D* *552D* *553D* *554D* *555D* *556D* *557D* *558D* *559D* *560D* *561D* *562D* *563D* *564D* *565D* *566D* *567D* *568D* *569D* *570D* *571D* *572D* *573D* *574D* *575D* *576D* *577D* *578D* *579D* *580D* *581D* *582D* *583D* *584D* *585D* *586D* *587D* *588D* *589D* *590D* *591D* *592D* *593D* *594D* *595D* *596D* *597D* *598D* *599D* *600D* *601D* *602D* *603D* *604D* *605D* *606D* *607D* *608D* *609D* *610D* *611D* *612D* *613D* *614D* *615D* *616D* *617D* *618D* *619D* *620D* *621D* *622D* *623D* *624D* *625D* *626D* *627D* *628D* *629D* *630D* *631D* *632D* *633D* *634D* *635D* *636D* *637D* *638D* *639D* *640D* *641D* *642D* *643D* *644D* *645D* *646D* *647D* *648D* *649D* *650D* *651D* *652D* *653D* *654D* *655D* *656D* *657D* *658D* *659D* *660D* *661D* *662D* *663D* *664D* *665D* *666D* *667D* *668D* *669D* *670D* *671D* *672D* *673D* *674D* *675D* *676D* *677D* *678D* *679D* *680D* *681D* *682D* *683D* *684D* *685D* *686D* *687D* *688D* *689D* *690D* *691D* *692D* *693D* *694D* *695D* *696D* *697D* *698D* *699D* *700D* *701D* *702D* *703D* *704D* *705D* *706D* *707D* *708D* *709D* *710D* *711D* *712D* *713D* *714D* *715D* *716D* *717D* *718D* *719D* *720D* *721D* *722D* *723D* *724D* *725D* *726D* *727D* *728D* *729D* *730D* *731D* *732D* *733D* *734D* *735D* *736D* *737D* *738D* *739D* *740D* *741D* *742D* *743D* *744D* *745D* *746D* *747D* *748D* *749D* *750D* *751D* *752D* *753D* *754D* *755D* *756D* *757D* *758D* *759D* *760D* *761D* *762D* *763D* *764D* *765D* *766D* *767D* *768D* *769D* *770D* *771D* *772D* *773D* *774D* *775D* *776D* *777D* *778D* *779D* *780D* *781D* *782D* *783D* *784D* *785D* *786D* *787D* *788D* *789D* *790D* *791D* *792D* *793D* *794D* *795D* *796D* *797D* *798D* *799D* *800D* *801D* *802D* *803D* *804D* *805D* *806D* *807D* *808D* *809D* *810D* *811D* *812D* *813D* *814D* *815D* *816D* *817D* *818D* *819D* *820D* *821D* *822D* *823D* *824D* *825D* *826D* *827D* *828D* *829D* *830D* *831D* *832D* *833D* *834D* *835D* *836D* *837D* *838D* *839D* *840D* *841D* *842D* *843D* *844D* *845D* *846D* *847D* *848D* *849D* *850D* *851D* *852D* *853D* *854D* *855D* *856D* *857D* *858D* *859D* *860D* *861D* *862D* *863D* *864D* *865D* *866D* *867D* *868D* *869D* *870D* *871D* *872D* *873D* *874D* *875D* *876D* *877D* *878D* *879D* *880D* *881D* *882D* *883D* *884D* *885D* *886D* *887D* *888D* *889D* *890D* *891D* *892D* *893D* *894D* *895D* *896D* *897D* *898D* *899D* *900D* *901D* *902D* *903D* *904D* *905D* *906D* *907D* *908D* *909D* *910D* *911D* *912D* *913D* *914D* *915D* *916D* *917D* *918D* *919D* *920D* *921D* *922D* *923D* *924D* *925D* *926D* *927D* *928D* *929D* *930D* *931D* *932D* *933D* *934D* *935D* *936D* *937D* *938D* *939D* *940D* *941D* *942D* *943D* *944D* *945D* *946D* *947D* *948D* *949D* *950D* *951D* *952D* *953D* *954D* *955D* *956D* *957D* *958D* *959D* *960D* *961D* *962D* *963D* *964D* *965D* *966D* *967D* *968D* *969D* *970D* *971D* *972D* *973D* *974D* *975D* *976D* *977D* *978D* *979D* *980D* *981D* *982D* *983D* *984D* *985D* *986D* *987D* *988D* *989D* *990D* *991D* *992D* *993D* *994D* *995D* *996D* *997D* *998D* *999D* *1000D* *1001D* *1002D* *1003D* *1004D* *1005D* *1006D* *1007D* *1008D* *1009D* *1010D* *1011D* *1012D* *1013D* *1014D* *1015D* *1016D* *1017D* *1018D* *1019D* *1020D* *1021D* *1022D* *1023D* *1024D* *1025D* *1026D* *1027D* *1028D* *1029D* *1030D* *1031D* *1032D* *1033D* *1034D* *1035D* *1036D* *1037D* *1038D* *1039D* *1040D* *1041D* *1042D* *1043D* *1044D* *1045D* *1046D* *1047D* *1048D* *1049D* *1050D* *1051D* *1052D* *1053D* *1054D* *1055D* *1056D* *1057D* *1058D* *1059D* *1060D* *1061D* *1062D* *1063D* *1064D* *1065D* *1066D* *1067D* *1068D* *1069D* *1070D* *1071D* *1072D* *1073D* *1074D* *1075D* *1076D* *1077D* *1078D* *1079D* *1080D* *1081D* *1082D* *1083D* *1084D* *1085D* *1086D* *1087D* *1088D* *1089D* *1090D* *1091D* *1092D* *1093D* *1094D* *1095D* *1096D* *1097D* *1098D* *1099D* *1100D* *1101D* *1102D* *1103D* *1104D* *1105D* *1106D* *1107D* *1108D* *1109D* *1110D* *1111D* *1112D* *1113D* *1114D* *1115D* *1116D* *1117D* *1118D* *1119D* *1120D* *1121D* *1122D* *1123D* *1124D* *1125D* *1126D* *1127D* *1128D* *1129D* *1130D* *1131D* *1132D* *1133D* *1134D* *1135D* *1136D* *1137D* *1138D* *1139D* *1140D* *1141D* *1142D* *1143D* *1144D* *1145D* *1146D* *1147D* *1148D* *1149D* *1150D* *1151D* *1152D* *1153D* *1154D* *1155D* *1156D* *1157D* *1158D* *1159D* *1160D* *1161D* *1162D* *1163D* *1164D* *1165D* *1166D* *1167D* *1168D* *1169D* *1170D* *1171D* *1172D* *1173D* *1174D* *1175D* *1176D* *1177D* *1178D* *1179D* *1180D* *1181D* *1182D* *1183D* *1184D* *1185D* *1186D* *1187D* *1188D* *1189D* *1190D* *1191D* *1192D* *1193D* *1194D* *1195D* *1196D* *1197D* *1198D* *1199D* *1200D* *1201D* *1202D* *1203D* <

Quadric Surfaces

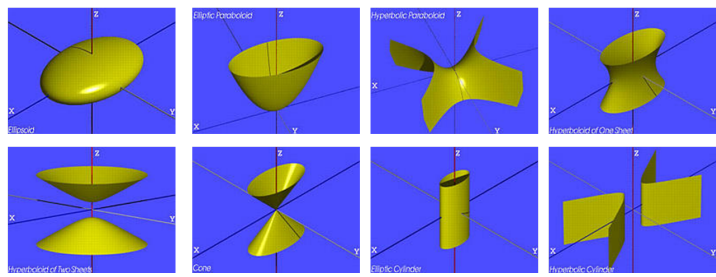
- 곡면을 표현하는 방법 중 하나
- 계수를 어떻게 고르는 지에 따라 여러가지 형태의 곡면이 나옴



(문제점) 내가 원하는 shape를 만들기 위해 계수들을 정해야 하는데, shape와 계수가 직관적으로 연결되지 않음  
→ 어떤 식으로 바뀔지 사용자 입장에서 불분명 → 사용성 떨어짐

$$\sum_{i,j=1}^D Q_{i,j} x_i x_j + \sum_{i=1}^D P_i x_i + R = 0$$

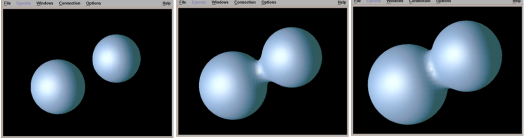
$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0.$$

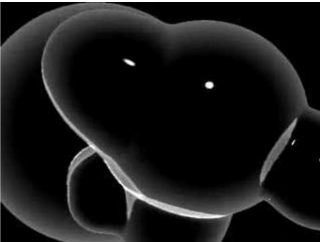


Bloppy Object

- 곡면을 표현하는 방법 중 하나
- Non-rigid objects 를 모델링
  - 고무, 분자, 액체, 물방울
  - 떨어져 있을 때는 독립적인 구 형태이다가, 가까워지면 연결되며 하나로 붙음

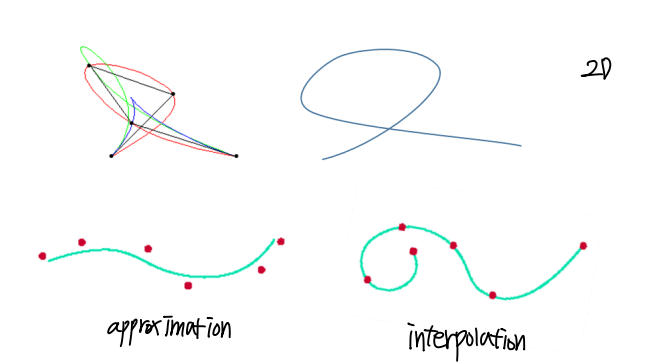
$$f(x, y, z) = \sum_k b_k e^{-a_k r_k^2} - T = 0 \quad r_k^2 = x_k^2 + y_k^2 + z_k^2$$







Splines

- 정의
  - 일반적인 곡선, 곡면을 모델링할 때 사용
  - Control points 로 정의
  - Spline curve 의 차수 = Control points 의 개수 - 1
    - ex. 3개의 Control points 를 사용하는 경우, 차수는 2차
- Approximation vs Interpolation
  - Approximation: Spline curve 가 모든 Control Points 를 통과하지는 않고, 근사
  - Interpolation: Spline curve 가 모든 Control Points를 정확히 통과
- Piecewise construction
  - 하나의 복잡한 curve 를 만들기 위해, 차수가 낮은(대부분 3 이하) 여러 curve 를 연결하는 방식



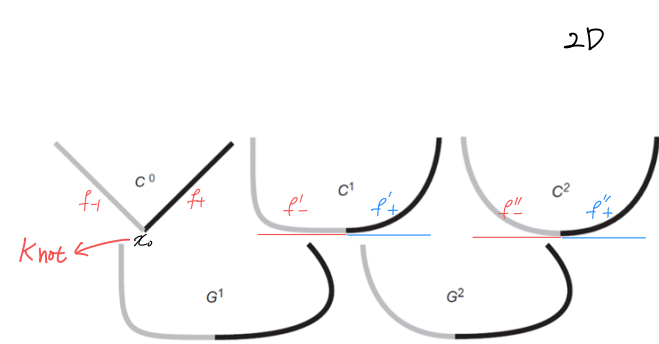


사용하는 이유  
⇒ 높은 차수 다항식의 문제점: 높은 차수의 다항식은 곡선을 더 복잡하게 모델링할 수 있지만, 계산 중 오차가 누적될 가능성이 높음



두 curve 가 자연스럽게 연결되어야 하나의 curve 같은 효과를 줄 수 있음  
→ Continuity 가 중요

- Continuity (C<sup>n</sup> vs G<sup>n</sup>)
  - C<sup>n</sup> : 왼쪽 curve 와 오른쪽 curve 각각 n번 미분한 값이 동일
    - C<sup>0</sup>: curve 의 끝점들이 정확히 일치
    - C<sup>1</sup>: curve 를 1번 미분한 값(접선)이 동일 → Tangent continuous
    - C<sup>2</sup>: curve 를 2번 미분한 값(곡률)이 동일 → Curvature continuous
    - n 이 클수록 smoothe 하게 연결
  - G<sup>n</sup> : 왼쪽 curve 와 오른쪽 curve 각각 n-1번 미분한 값이 동일(C<sup>n</sup>-1) + n번 미분한 값이 배수 관계 → 방향성 같다
- ⇒ 보통 C<sup>2</sup>, G<sup>2</sup> 사용
- Bezier, b-spline, NURBS
  - 모두 다항 함수이지만, 표현하기 위한 Basis function 에 따라 종류가 달라짐



## Bezier Curves and Surfaces

### Bezier curve

- Spline curve 의 일종으로, **Bernstein Polynomials** 을 Basis function 으로 사용

$$\mathbf{p}(u) = \sum_{k=0}^n \mathbf{p}_k b_{k,n}(u), \quad 0 \leq u \leq 1$$

- n: Curve 의 차수,  $\mathbf{p}_k$ : Control point,  $\mathbf{b}_{k,n}(u)$ : n차 **Bernstein Polynomials**,  $\mathbf{P}(u)$ : **Bezier curve**
  - n차 Bezier curve 는 n+1개의 Control point 로 정의
  - $\mathbf{P}(u)$  는  $(x(u), y(u)) \rightarrow$  2차원 Bezier curve 위의 점

### Bernstein Polynomials 특성

$$b_{k,n}(u) = {}_nC_k u^k (1-u)^{n-k}, \quad {}_nC_k = \frac{n!}{k!(n-k)!}, \quad 0 \leq k \leq n$$

$$\begin{array}{l} 0\text{차} \\ 1\text{차} \end{array} \quad \begin{array}{l} \{b_{0,0}(u) = 1\} \\ \begin{cases} b_{0,1}(u) = 1-u \\ b_{1,1}(u) = u \end{cases} \end{array} \quad \begin{array}{l} \begin{cases} b_{0,2}(u) = (1-u)^2 \\ b_{1,2}(u) = 2(1-u)u \\ b_{2,2}(u) = u^2 \end{cases} \\ \begin{cases} b_{0,3}(u) = (1-u)^3 \\ b_{1,3}(u) = 3(1-u)^2u \\ b_{2,3}(u) = 3(1-u)u^2 \\ b_{3,3}(u) = u^3 \end{cases} \end{array}$$

$$\sum_{k=0}^n b_{k,n}(u) = 1$$

- 모든  $u$  에 대해  $\mathbf{b}_{k,n}(u)$  은 항상 양수 ( $\rightarrow$  Curve 가 control point 의 가중 평균 형태를 가지게 함)
- 합이 1 ( $\rightarrow$  Curve 의 전체 합이 일정하게 유지)



$\Rightarrow$  이 두 특성으로 인해, **Bezier curve** 는 **Convex hull property** 를 가짐

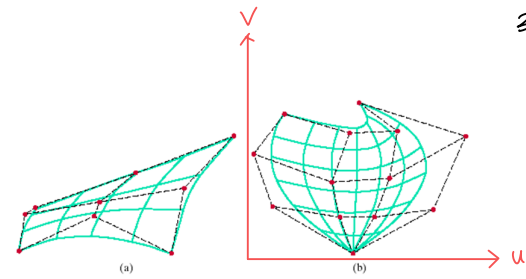
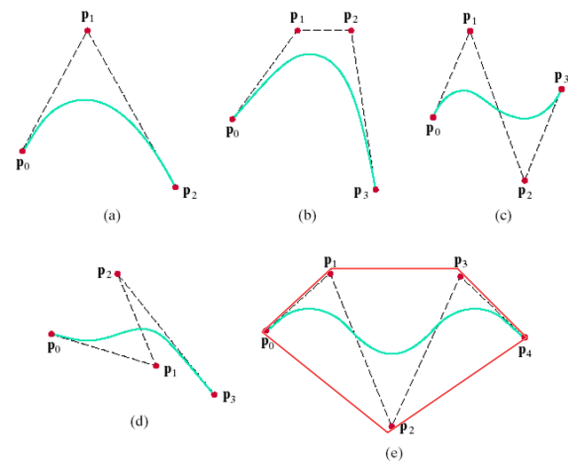


### Convex hull property

- Convex hull: Control points 를 모두 포함하는 최소한의 볼록 다각형
- Bezier curve  $\mathbf{P}(u)$  는 절대로 Convex hull 바깥으로 나가지 않음  
 $\rightarrow$  Bezier curve 는 Convex hull 안에 포함되므로, Bounding volume 으로 사용 가능

### Surface의 경우

- $u \rightarrow (\mathbf{u}, \mathbf{v})$  로 확장 ( $0 \leq u \leq 1, 0 \leq v \leq 1$ )



3D

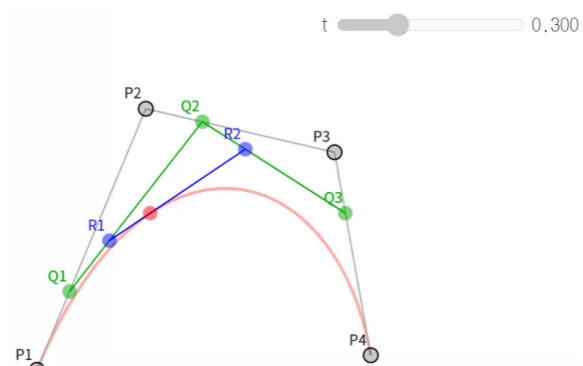
## De Casteljau's algorithm

- Bezier curve 를 그리는 기하학적 방법
- $\mathbf{P}(u)$  를 다항식으로 직접 계산하지 않고, **Recursive 한 Linear interpolation** 을 사용
- 방법
  - Recursive 하게 매개변수  $t$  에 따라 **(1-t) : t 비율로** 내분점을 찾음 ( $0 \leq t \leq 1$ )
  - $t=0$  이면 Curve 의 시작점  $\mathbf{P}_1$ ,  $t=1$  이면 끝점  $\mathbf{P}_4$



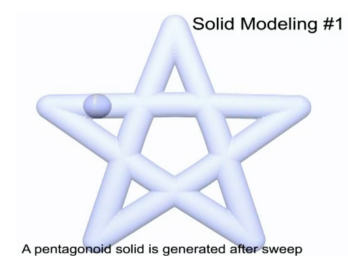
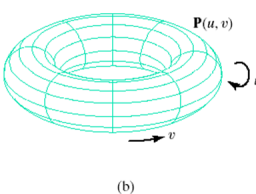
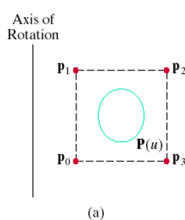
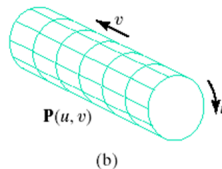
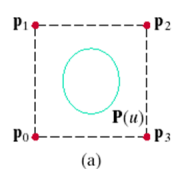
$t=0.3$  일 때, Control points 개수 4개( $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4$ )  $\rightarrow$  3차 Bezier curve 를 그려보자

- $\mathbf{P}_1\text{-}\mathbf{P}_2, \mathbf{P}_2\text{-}\mathbf{P}_3, \mathbf{P}_3\text{-}\mathbf{P}_4$  1:3 내분점 찾기  $\rightarrow \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3$
- $\mathbf{Q}_1\text{-}\mathbf{Q}_2, \mathbf{Q}_2\text{-}\mathbf{Q}_3$  1:3 내분점 찾기  $\rightarrow \mathbf{R}_1, \mathbf{R}_2$
- $\mathbf{R}_1\text{-}\mathbf{R}_2$  1:3 내분점 찾기  $\rightarrow$  Curve 상의 한 점

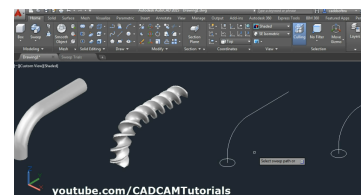


## Sweep Representation

- Profile curve (간단한 2D 형태) 를 정의
- Profile curve 를 어느 방향으로 Sweep 할 것인지 지정
  - 원을 직선 방향으로 sweep  $\rightarrow$  실린더
  - 원을 축을 기준으로 회전시키며 sweep  $\rightarrow$  도넛 모양



A pentagonoid solid is generated after sweep

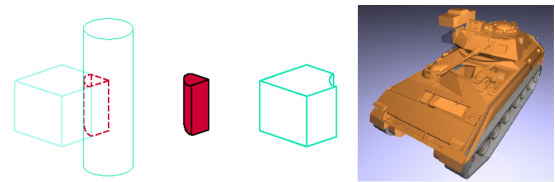


Constructive Solid Geometry

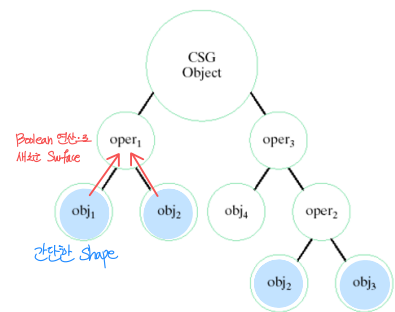
- 복잡한 Shape 를 만들기 위해, 간단한 Shape 들에 **Boolean operation(Union, Intersection, Difference)** 을 하는 방법

💡 실제로는 매우 많은 Boolean operation 이 필요 → CSG tree 사용

- **CSG tree**
  - Leaf node: 간단한 Shape
  - Internal Node: Boolean operation
- 동작 방식:
  - CSG tree 의 Leaf node 에 있는 간단한 Shape 들을 Boolean operation 을 통해 조합하여 최종 결과물을 생성



8456 CSG operations



Quiz) de Casteljau's Algorithm

Given the following cubic Bezier curve  $\mathbf{P}(u)$  in 2D in Eq1, we want to find the point  $\mathbf{P}(\frac{1}{2})$  on the curve:

Eq1:  $\mathbf{P}(u) = \sum_{k=0}^3 \mathbf{P}_k b_k(u)$ , where  $b_k(u) = \frac{3!}{k!(3-k)!} u^k (1-u)^{(3-k)}$

with control points  $\mathbf{P}_0 = (-8, 4)$ ,  $\mathbf{P}_1 = (-4, 4)$ ,  $\mathbf{P}_2 = (0, 0)$ ,  $\mathbf{P}_3 = (4, -4)$ .

- Fill the blanks in the following with a decimal number, not a fractional number (e.g. 0.5 instead of 1/2 or  $\frac{1}{2}$ ).
- If you click <체크>, your answer will be verified and scored immediately.

문제 1

정답

총 4.00 점에서  
4.00 점 획득

🗑 문제 표시

Find  $\mathbf{P}(\frac{1}{2}) = ( \text{ } -2 \text{ } \checkmark , \text{ } 1.5 \text{ } \checkmark )$  by directly evaluating  $u = \frac{1}{2}$  in Eq1.

$\mathbf{P}(\frac{1}{2}) = \sum_{k=0}^3 \mathbf{P}_k \frac{3!}{k!(3-k)!} (\frac{1}{2})^k (1 - \frac{1}{2})^{(3-k)}$

$\mathbf{P}(\frac{1}{2}) = (-2, 1.5)$ .

$$\begin{aligned} \mathbf{P}(\frac{1}{2}) &= \sum_{k=0}^3 \mathbf{P}_k \cdot \frac{3!}{k!(3-k)!} \times (\frac{1}{2})^k (\frac{1}{2})^{3-k} \\ &= \mathbf{P}_0 \times \frac{3!}{0!3!} \times (\frac{1}{2})^0 (\frac{1}{2})^3 \rightarrow (-8, 4) \times \frac{1}{8} \\ &\quad + \mathbf{P}_1 \times \frac{3!}{1!2!} \times (\frac{1}{2})^1 (\frac{1}{2})^2 \rightarrow (-4, 4) \times \frac{3}{8} \times \frac{1}{4} \\ &\quad + \mathbf{P}_2 \times \frac{3!}{2!1!} \times (\frac{1}{2})^2 (\frac{1}{2})^1 \rightarrow (0, 0) \times \frac{3}{4} \times \frac{1}{2} \\ &\quad + \mathbf{P}_3 \times \frac{3!}{3!0!} \times (\frac{1}{2})^3 (\frac{1}{2})^0 \rightarrow (4, -4) \times \frac{1}{8} \\ &= \frac{1}{8} \{ (-8, 4) + (-12, 12) + (0, 0) + (4, -4) \} \\ &= \frac{1}{8} (-8, 12) = (-1, \frac{3}{2}) \end{aligned}$$

정보

🗑 문제 표시

We want to evaluate  $\mathbf{P}(\frac{1}{2})$  by using de Casteljau's algorithm.

문제 2

정답

총 6.00 점에서  
6.00 점 획득

🗑 문제 표시

The first three points generated by de Casteljau's algorithm using  $\overline{\mathbf{P}_0 \mathbf{P}_1}, \overline{\mathbf{P}_1 \mathbf{P}_2}, \overline{\mathbf{P}_2 \mathbf{P}_3}$  are  $\mathbf{P}'_0 = ( \text{ } -6 \text{ } \checkmark , \text{ } 4 \text{ } \checkmark )$ ,  $\mathbf{P}'_1 = ( \text{ } -2 \text{ } \checkmark , \text{ } 2 \text{ } \checkmark )$ ,  $\mathbf{P}'_2 = ( \text{ } 2 \text{ } \checkmark , \text{ } -2 \text{ } \checkmark )$ .

$\mathbf{P}'_0 = \frac{1}{2}(\mathbf{P}_0 + \mathbf{P}_1)$   
 $\mathbf{P}'_1 = \frac{1}{2}(\mathbf{P}_1 + \mathbf{P}_2)$   
 $\mathbf{P}'_2 = \frac{1}{2}(\mathbf{P}_2 + \mathbf{P}_3)$   
 $\mathbf{P}'_0 = (-6, 4)$ ,  $\mathbf{P}'_1 = (-2, 2)$ ,  $\mathbf{P}'_2 = (2, -2)$ .

$\mathbf{P}(\frac{1}{2})$  이면  $i=1$  번째.

$$\begin{aligned} &\begin{matrix} \mathbf{P}_0 & & \mathbf{P}_1 & & \mathbf{P}_2 & & \mathbf{P}_3 \\ (-8, 4) & & (-4, 4) & & (0, 0) & & (4, -4) \end{matrix} \\ &\begin{matrix} \swarrow & & \swarrow & & \swarrow \\ \frac{(-8, 4) + (-4, 4)}{2} & & \frac{(-4, 4) + (0, 0)}{2} & & \frac{(0, 0) + (4, -4)}{2} \\ = (-6, 4) & & = (-2, 2) & & = (2, -2) \end{matrix} \\ &\begin{matrix} \swarrow & & \swarrow \\ \frac{(-6, 4) + (-2, 2)}{2} & & \frac{(-2, 2) + (2, -2)}{2} \\ = (-4, 3) & & = (0, 0) \end{matrix} \\ &\begin{matrix} \swarrow \\ \frac{(-4, 3) + (0, 0)}{2} \\ = (-2, \frac{3}{2}) \end{matrix} \end{aligned}$$

문제 3

정답

총 4.00 점에서  
4.00 점 획득

🗑 문제 표시

The second two points generated by de Casteljau's algorithm using  $\overline{\mathbf{P}'_0 \mathbf{P}'_1}, \overline{\mathbf{P}'_1 \mathbf{P}'_2}$  are  $\mathbf{P}''_0 = ( \text{ } -4 \text{ } \checkmark , \text{ } 3 \text{ } \checkmark )$ ,  $\mathbf{P}''_1 = ( \text{ } 0 \text{ } \checkmark , \text{ } 0 \text{ } \checkmark )$ .

$\mathbf{P}''_0 = \frac{1}{2}(\mathbf{P}'_0 + \mathbf{P}'_1)$   
 $\mathbf{P}''_1 = \frac{1}{2}(\mathbf{P}'_1 + \mathbf{P}'_2)$   
 $\mathbf{P}''_0 = (-4, 3)$ ,  $\mathbf{P}''_1 = (0, 0)$ .

문제 4

정답

총 2.00 점에서  
2.00 점 획득

🗑 문제 표시

The final point generated by de Casteljau's algorithm using  $\overline{\mathbf{P}''_0 \mathbf{P}''_1}$  is  $\mathbf{P}(\frac{1}{2}) = ( \text{ } -2 \text{ } \checkmark , \text{ } 1.5 \text{ } \checkmark )$ .

$\mathbf{P}(\frac{1}{2}) = \frac{1}{2}(\mathbf{P}''_0 + \mathbf{P}''_1)$   
 $\mathbf{P}(\frac{1}{2}) = (-2, 1.5)$ .