

Framework computacional en Python para la construcción de superficies tipo barrido a partir de curvas NURBS

Santamaria Ronald¹; Castillo Yheff¹; Urbina Rubén¹; Jimenez Judith¹

¹Universidad Nacional de Piura, Perú, 72937144@posgrado.unp.edu.pe, ycastillom@egresados.unp.edu.pe, rurbinag@unp.edu.pe, jjimenezv@unp.edu.pe

Resumen– Este trabajo presenta un framework para generar superficies de barrido (regladas, cilíndricas y cilíndricas por un escalor) basado exclusivamente en curvas NURBS a través de puntos de control, eliminando la dependencia de fórmulas paramétricas analíticas. La propuesta permite construir superficies mediante transformaciones directas entre curvas generatrices y trayectorias, introduciendo un modelo de "cilindro con escalor" para controlar la expansión de la sección transversal. El enfoque garantiza un control geométrico local preciso y una estructura computacional unificada para el diseño asistido por computadora.

Palabras clave: NURBS, Modelado geométrico, Superficies de barrido, Puntos de control, Geometría computacional.

I. INTRODUCCIÓN

El modelado geométrico de superficies a partir de curvas constituye un problema central en la geometría computacional y el diseño asistido por computadora (CAGD), con aplicaciones directas en ingeniería, arquitectura, manufactura y visualización científica. Superficies como, las superficies regladas, cilíndricas y de barrido representan herramientas fundamentales para la generación de geometrías complejas a partir de entidades curvas simples, permitiendo describir formas tridimensionales mediante procesos constructivos bien definidos. Tradicionalmente, estas superficies se formulan mediante expresiones paramétricas analíticas que describen explícitamente la dependencia funcional entre los parámetros y las coordenadas espaciales [1, 2, 3, 4].

Sin embargo, en entornos computacionales orientados al diseño interactivo, las formulaciones paramétricas clásicas presentan limitaciones importantes. En la práctica, el diseño geométrico no se realiza manipulando funciones continuas, sino mediante estructuras discretas, principalmente puntos de control, pesos y vectores de nudos. Este hecho ha motivado el uso extensivo de curvas y superficies NURBS, las cuales ofrecen una representación flexible, precisa y compatible con sistemas CAD modernos. A pesar de ello, muchos enfoques continúan construyendo superficies evaluando primero curvas paramétricas y aplicando posteriormente transformaciones geométricas, lo que introduce una separación conceptual entre la teoría continua y la implementación computacional [5].

En el contexto específico de superficies de barrido, una línea de investigación relevante se centra en la correcta orientación del perfil durante el desplazamiento a lo largo de

una trayectoria espacial. En este marco, se han propuesto métodos basados en rotation minimizing frames, los cuales buscan evitar rotaciones indeseadas del perfil y mejorar la suavidad de la superficie generada. Un ejemplo representativo es el trabajo sobre el cálculo robusto de marcos de rotación mínima, donde se propone aproximar la trayectoria mediante arcos circulares y generar superficies de barrido suaves y cerradas, preservando la estructura NURBS cuando el perfil es una curva NURBS. Estos enfoques abordan de manera eficaz el problema de la orientación diferencial del perfil, pero dependen explícitamente del cálculo de marcos móviles y de aproximaciones geométricas de la trayectoria, lo que añade complejidad al proceso de construcción [6].

Por otro lado, existen trabajos orientados principalmente a la manufactura, como aquellos dedicados al mecanizado NC de superficies de revolución y superficies barridas con perfiles libres formulados mediante curvas de Bézier [7]. En estos estudios, el énfasis se sitúa en la precisión geométrica, la simplicidad computacional y la aplicabilidad industrial, particularmente en el diseño de moldes y herramientas. Si bien estos enfoques son altamente relevantes en el contexto CAM, su objetivo no es la formulación general de un modelo geométrico flexible basado en NURBS, sino la generación eficiente de trayectorias de mecanizado a partir de perfiles predefinidos.

En contraste con estos trabajos, el presente artículo propone un framework computacional para la construcción de superficies tipo barrido basado exclusivamente en puntos de control NURBS, sin recurrir a formulaciones paramétricas analíticas explícitas ni al uso de marcos móviles. La contribución principal consiste en reformular la generación de superficies regladas, cilíndricas y de barrido como un proceso directo sobre estructuras NURBS discretas, donde las superficies se construyen a nivel de puntos de control mediante combinaciones afines y transformaciones controladas [8, 9]. Adicionalmente, se introduce una extensión de las superficies cilíndricas mediante un factor escalar dependiente del parámetro transversal, lo que permite modelar contracciones y expansiones progresivas de la sección transversal durante la traslación.

De este modo, este trabajo propone como una alternativa complementaria a los enfoques basados en marcos de referencia y a otros métodos [10], ofreciendo una estrategia unificada, extensible y computacionalmente coherente para el modelado

de superficies NURBS desde puntos de control. El enfoque propuesto resulta especialmente adecuado para aplicaciones de diseño geométrico asistido por computadora y visualización científica, donde el control geométrico local y la claridad estructural del modelo son aspectos prioritarios.

II. PRELIMINARES MATEMÁTICOS

Se presentan a continuación las definiciones basadas en [1]–[5].

A. Curvas NURBS

Una curva NURBS de p -grado está definida por:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i P_i}{\sum_{i=0}^n N_{i,p}(u) w_i}, \quad a \leq u \leq b \quad (1)$$

Donde:

- Los P_i son los puntos de control (formando un polígono de control). Estos influyen en la forma de la curva y no necesariamente todos los puntos deben pertenecer a la curva.
- Los w_i son los pesos, que se asocian a cada punto de control P_i .
- Las $N_{i,p}(u)$ son las funciones base de B-spline de grado p definidas en un vector de nudos no periódico $U = \{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b, \dots, b\}$.

A menos que se indique lo contrario, asumimos que $a = 0$, $b = 1$, y que $w_i > 0$ para todo i .

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^n N_{j,p}(u) w_j} \quad (2)$$

Nos permite reescribir la ecuación en la forma:

$$C(u) = \sum_{i=0}^n R_{i,p}(u) P_i \quad (3)$$

Las $R_{i,p}(u)$ son las funciones base racionales; son funciones racionales definidas por partes en $u \in [0,1]$.

B. Superficies NURBS

Una superficie NURBS de grado p en la dirección u y grado q en la dirección v es una función racional por partes bivariada y vectorial de la forma:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}}, \quad 0 \leq u, v \leq 1 \quad (4)$$

Los $P_{i,j}$ forman una red de control bidireccional, los $w_{i,j}$ son los pesos, y las $N_{i,p}(u)$ y $N_{j,q}(v)$ son las funciones base de B-spline no racionales definidas en los vectores de nudos.

$$U = \{0, \dots, 0, u_{p+1}, \dots, u_{r-p-1}, 1, \dots, 1\}, \quad p+1 \quad (5)$$

$$V = \{0, \dots, 0, v_{q+1}, \dots, v_{s-q-1}, 1, \dots, 1\}, \quad q+1 \quad (6)$$

Donde $r = n + p + 1$ y $s = m + q + 1$.

Introduciendo las funciones base racionales definidas por partes.

$$R_{i,j}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u) N_{l,q}(v) w_{k,l}} \quad (7)$$

Entonces, la ecuación de la superficie puede escribirse de la siguiente forma:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{i,j} \quad (8)$$

C. Superficies regladas

En el contexto de modelado geométrico, podemos definir una superficie reglada como el resultado de interpolar linealmente puntos correspondientes a un mismo valor del parámetro $u = \bar{u}$ situados sobre dos curvas $C_1(u)$, $C_2(u)$.

Estas superficies, frecuentemente utilizadas en la industria aeronáutica (las alas de los aviones son generalmente de este tipo), son simplemente generalizaciones de la superficie bilineal. En efecto, tendremos:

$$S(u, v) = (1 - v) C_1(u) + v C_2(u) \quad (9)$$

donde ahora $C_1(u)$, $C_2(u)$ pueden ser cualquiera curva 3D (la superficie bilineal corresponde al caso particular en el que $C_1(u)$ y $C_2(u)$ son rectas).

Evidentemente, las curvas isoparamétricas para $u = \text{constante}$ son rectas, mientras que para $v = \text{constante}$ son una combinación convexa de las curvas $C_1(u)$ y $C_2(u)$.

D. Superficies cilíndricas

Como casos particulares de superficies regladas están las denominadas cilindros generalizados, que se obtienen al desplazar una curva $C(u)$ a través del espacio siguiendo una trayectoria recta.

Llamando D al vector que indica la dirección del desplazamiento, y sustituyendo en (9) la $C_1(u)$ por la curva dada $C(u)$ y $C_2(u)$ por la curva trasladada $C(u) + D$, se tiene

$$S(u, v) = (1 - v)C(u) + v(C(u) + D) = C(u) + vD \quad (10)$$

Con frecuencia la dirección D se elige perpendicular al plano que contiene la curva $C(u)$ si ésta es plana, en cuyo caso tendríamos un cilindro recto.

Es interesante destacar que las superficies cilíndricas en general son desarrollables, significando esto que pueden ser deformadas en un plano sin variar las distancias entre cualesquiera de sus puntos. Recíprocamente, y esto es realmente importante por ejemplo en la industria metalúrgica, una superficie desarrollable (en particular cilíndrica) puede ser obtenida a partir de un plano mediante rotaciones infinitesimales sin estirarla ni contraerla.

E. Superficies generadas por barrido

Aunque los tipos de superficies vistos en las secciones anteriores tienen su propia razón de ser y constituyen buenas herramientas constructivas, son sólo casos particulares de una técnica de generación de superficies denominada barrido o *sweeping*.

La idea, ya explotada para definir las superficies cilíndricas y las de revolución, consiste en desplazar una curva $C(u)$ a través del espacio, barriéndolo, mientras se le obliga a seguir una trayectoria $T(v)$. Más general aún es permitir que la curva pueda transformarse mediante rotaciones y escalados a la vez que se mueve.

La ecuación para esta clase de superficies es $S(u, v) = A(v)C(u) + T(v)$, donde $A(v)$ es una afinidad (sin traslación) que en la práctica se toma como combinación de una rotación y un escalado.

Observemos que la curva $C(u)$ y la trayectoria $T(v)$ no tienen por qué tener algún punto en común, por lo que en general la superficie no contendrá a la curva trayectoria.

Como caso particular interesante tenemos las denominadas superficies de barrido traslacional, que corresponde a tomar $A(v) = I$ (matriz identidad), en cuyo caso $S(u, v) = C(u) + T(v)$.

III. GENERALIZACIÓN CON NURBS

A. Superficie Reglada

En esta extensión, la construcción de superficies regladas no se realiza a partir de expresiones paramétricas analíticas, sino exclusivamente mediante puntos de control que definen curvas NURBS, reforzando así el control geométrico local y la flexibilidad del modelado.

Recordemos que una superficie reglada clásica puede expresarse como una interpolación lineal entre dos curvas generatrices $C_1(u)$ y $C_2(u)$, dada por

$$S(u, v) = C_1(u) + v(C_2(u) - C_1(u)) \quad (11)$$

En la implementación propuesta, las curvas $C_1(u)$ y $C_2(u)$ no se describen mediante funciones explícitas, sino a través de listas de puntos de control que definen curvas NURBS de igual grado y número de puntos. Esta condición es necesaria para garantizar una interpolación coherente entre curvas correspondientes.

A partir de esto, se consideran tres conjuntos de puntos de control:

- pts_1 : puntos de control de la curva $C_1(u)$, contenida en el plano XZ;
- pts_3 : puntos de control de la curva $C_2(u)$, también paralela en el plano XZ;
- pts_2 : puntos de control asociados al parámetro v , que en este caso describen una recta contenida en el plano YZ y actúan como parámetro de interpolación transversal.

```
class Nurbs_surface_ruler(Nurbs_surface_base):
    def __init__(self, pts1, pts2, pts3, *args):
        super().__init__(pts1, pts2, *args)
        self.pts3 = np.array(pts3, dtype=float)

    def P_ij(self, i, j):
        return [
            self.pts1[i][0] +
            self.pts2[j][1]*(self.pts3[i][0]-self.pts1[i][0]),
            self.pts2[j][1]*self.pts3[i][1], self.pts1[i][2] +
            self.pts2[j][1]*(self.pts3[i][2]-self.pts1[i][2]))]
```

De este modo, la superficie reglada se construye directamente a nivel de puntos de control de la superficie, evitando el paso intermedio por evaluaciones funcionales de las curvas. Cada punto de control de la superficie, denotado por $P_{i,j}$, se obtiene mediante una interpolación lineal entre los puntos de control correspondientes de las curvas generatrices, modulada por el parámetro v .

Este enfoque permite reinterpretar la superficie reglada como una estructura tensorial de puntos de control, donde la dirección u está gobernada por la definición NURBS de las curvas generatrices, y la dirección v controla la interpolación lineal entre ellas.

En la siguiente superficie reglada NURBS se construye a partir de dos curvas generatrices C_1 y C_2 , definidas exclusivamente mediante puntos de control, pesos y vectores de nudos. En el ejemplo considerado, ambas curvas poseen el mismo número de puntos de control, el mismo grado y un vector de nudos idéntico, condición necesaria para garantizar la correspondencia entre puntos homólogos y permitir una interpolación coherente. La dirección reglada se define mediante una curva lineal independiente, caracterizada también por sus puntos de control, pesos y nudos, que actúa como parámetro transversal de interpolación.

```
# Construimos los datos necesarios
C_gen1 = [[2,0,1], [2,0,2.7321], [0.5,0,1.8660], [-1,0,1],
[0.5,0,0.1340], [2,0,-0.7321], [2,0,1]]
peso1 = [1, 1/2, 1, 1/2, 1, 1/2, 1]
nodo1 = [0, 0, 0, 1, 1, 2, 2, 3, 3, 3]
C_dir = [[0, 0, 0], [0, 1, 0]]
peso2 = [1, 1]
nodo2 = [0, 0, 1, 1]
C_gen2 = [[3,3,1], [3,3,2.7321], [0,3,1.8660], [-3,3,1],
[0,3,0.1340], [3,3,-0.7321], [3,3,1]]
```

Una vez definidos estos datos, la superficie se construye directamente a nivel de puntos de control mediante una combinación afín entre los puntos correspondientes de C_1 y C_2 , modulada por el parámetro transversal. El resultado es una superficie reglada suavemente mallada, limitada por dos curvas NURBS geoméricamente distintas, en este caso una circunferencia y una elipse, lo que evidencia la flexibilidad y el control geométrico local del método propuesto.

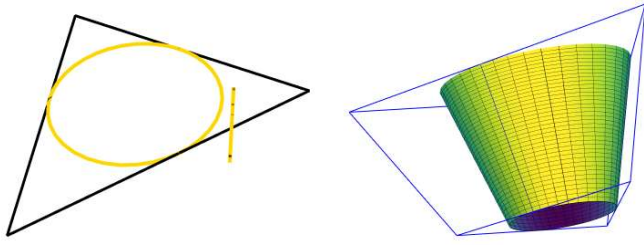


Fig 1. Superficie reglada NURBS delimitada por una circunferencia y una elipse.

B. Superficie por barrido

Las superficies de barrido pueden describirse mediante la expresión general $S(u, v) = A(v)C(u) + T(v)$, donde $C(u)$ es la curva generatriz, $T(v)$ es la trayectoria de barrido y $A(v)$ representa una transformación afín. En el caso particular de superficies de traslación, $A(v)$ corresponde a la identidad, mientras que el escalado puede incorporarse mediante una función escalar $r(v)$. En la metodología propuesta, tanto la curva generatriz como la trayectoria se definen exclusivamente mediante puntos de control, pesos y vectores de nudos, siguiendo la formulación NURBS.

Para el ejemplo considerado, la curva generatriz se define en el plano XZ y la trayectoria de barrido en el plano YZ . La superficie se construye directamente a nivel de puntos de control, combinando las estructuras NURBS de ambas curvas sin recurrir a expresiones paramétricas analíticas. Este enfoque permite generar superficies de barrido suavemente malladas con un control geométrico local preciso, evidenciando la flexibilidad del método para modelar superficies complejas a partir de curvas NURBS arbitrarias.

```
# Construimos los datos necesarios
C_gen = [[2,0,1], [2,0,2.7321], [0.5,0,1.8660], [-1,0,1],
         [0.5,0,0.1340], [2,0,-0.7321], [2,0,1]]
peso1 = [1, 1/2, 1, 1/2, 1, 1/2, 1]
nodo1 = [0, 0, 0, 1, 1, 2, 2, 3, 3]
C_dir = [[0, 0, 0], [0, 1.3879, 1.5809], [0, 3.2705, -
0.0252], [0, 5.0015, -1.5887], [0, 7.3201, 0.9533]]
peso2 = [1, 1, 1, 1, 1]
nodo2 = [0, 0, 0, 1/2, 1/2, 1, 1, 1]
# Inicializamos el objeto
surf = Nurbs_surface_cylindrical(C_gen, C_dir, peso1,
peso2, nodo1, nodo2, 2, 2)
# Visualizamos
Sweep1 = Nurbs_superficie3D_graph(surf)
Sweep1.grafica(pdir=False, pgen=False, c_pdir=True,
c_pgen=True, enmallar=True, pts_ctrl=True,
superficie=True)
```

Las siguientes superficies derivan de la superficie de barrido como casos particulares por lo cual su construcción es similar

C. Superficie cilíndrica

Si la curva generatriz se define como

$$C(u) = (f(u), 0, g(u)) \quad (12)$$

$$y \text{ la trayectoria de traslación como} \quad T(v) = (0, c(v), s(v)) \quad (13)$$

$$\text{la superficie cilíndrica resultante puede expresarse como} \quad S(u, v) = (f(u), c(v), g(u) + s(v)) \quad (14)$$

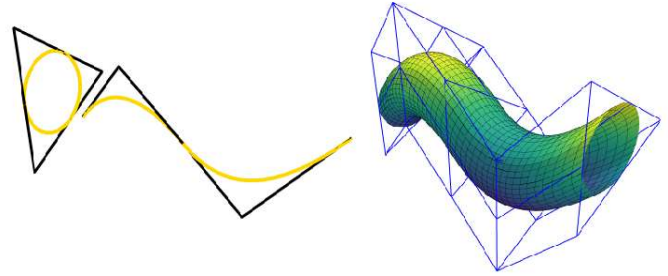


Fig 2. Superficie de barrido NURBS generada por la traslación de una circunferencia a lo largo de una trayectoria senoidal.

En el enfoque basado en NURBS, la curva generatriz $C(u)$ no se describe mediante funciones analíticas, sino a través de un conjunto de puntos de control, pesos y un vector de nudos. En la implementación, la lista `pts1` representa los puntos de control de la curva generatriz, mientras que la lista `pts2` corresponde a los puntos de control de la curva de traslación $T(v)$. De este modo, la superficie cilíndrica se construye directamente a nivel de puntos de control, donde cada punto $P_{i,j}$ de la superficie se obtiene combinando la coordenada transversal proveniente de la trayectoria con las coordenadas longitudinales de la curva generatriz.

```
class Nurbs_surface_cylindrical(Nurbs_surface_base):
    def P_ij(self, i, j):
        return [
            self.pts1[i][0],
            self.pts2[j][1],
            self.pts1[i][2] + self.pts2[j][2]
        ]
```

El siguiente ejemplo de una superficie en la forma senoidal.

```
# Construimos los datos
C_gen = [[0,0,0], [0,0,3], [3,0,3], [3,0,0], [0,0,0]]
peso1 = [1, 1, 1, 1, 1]
nodo1 = [0, 0, 1/4, 2/4, 3/4, 1, 1]
C_dir = [[0, 0, 0], [0, 1.3879, 1.5809], [0, 3.2705, -
0.0252], [0, 5.0015, -1.5887], [0, 7.3201, 0.9533]]
peso2 = [1, 1, 1, 1, 1]
nodo2 = [0, 0, 0, 1/2, 1/2, 1, 1, 1]

# Inicializamos el objeto
surf = Nurbs_surface_cylindrical(C_gen, C_dir, peso1,
peso2, nodo1, nodo2, 1, 2)
Sweep3 = Nurbs_superficie3D_graph(surf)
Sweep3.grafica(pdir=False, pgen=False, c_pdir=True,
c_pgen=True, enmallar=True, pts_ctrl=True,
superficie=True)
```

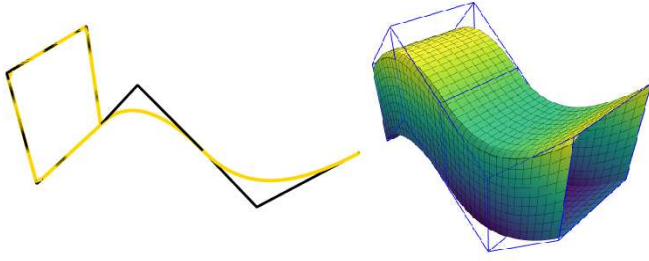


Fig 3. Superficie cilíndrica NURBS donde se traslada un cuadrado en forma senoidal

Otro ejemplo similar es el siguiente donde se utiliza una elipse en forma senoidal

```
# Construimos los datos
C_gen = [[3,0,1], [3,0,2.7321], [0,0,1.8660], [-3,0,1],
         [0,0,0.1340], [3,0,-0.7321], [3,0,1]]
peso1 = [1, 1/2, 1, 1/2, 1, 1/2, 1]
nodo1 = [0, 0, 0, 1, 1, 2, 2, 3, 3, 3]
C_dir = [[0, 0, 0], [0, 1.3879, 1.5809], [0, 3.2705, -
0.0252], [0, 5.0015, -1.5887], [0, 7.3201, 0.9533]]
peso2 = [1, 1, 1, 1, 1]
nodo2 = [0, 0, 0, 1/2, 1/2, 1, 1, 1]

# Inicializamos el objeto
surf = Nurbs_surface_cylindrical(C_gen, C_dir, peso1,
peso2, nodo1, nodo2, 2, 2)

# Visualizamos
Sweep2 = Nurbs_superficie3D_graph(surf)
Sweep2.grafica(pdir=False, pgen=False, c_pdir=True,
c_pgen=True, enmallar=True, pts_ctrl=True,
superficie=True)
```

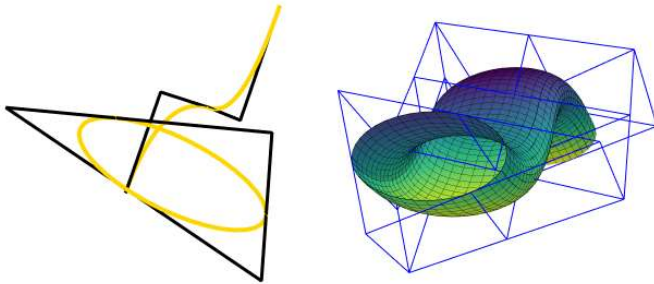


Fig 4. Superficie NURBS generada usando una elipse que se traslada en forma senoidal.

D. Superficie cilíndrica con un escalador

Las superficies cilíndricas con factor escalar constituyen una extensión de las superficies de traslación, en las que la transformación asociada al parámetro transversal incluye un escalamiento dependiente de v . Si la curva generatriz se define como $C(u) = (f(u), 0, g(u))$, contenida en el plano XZ , y la trayectoria de traslación como $T(v) = (0, c(v), s(v))$, la superficie resultante puede expresarse como

$$S(u, v) = (r(v)f(u), c(v), r(v)g(u) + s(v)) \quad (15)$$

donde $r(v)$ es una función escalar que controla el factor de dilatación de la curva generatriz a lo largo del barrido.

En el enfoque basado en NURBS, esta formulación se implementa sustituyendo las funciones analíticas por datos discretos. A partir de esto, la lista $pts1$ contiene los puntos de control de la curva generatriz $C(u)$, $pts2$ corresponde a los puntos de control de la trayectoria $T(v)$, y $pts3$ almacena los valores del factor escalar $r(v)$. A diferencia de los casos anteriores, $pts3$ no representa puntos geométricos en $\mathbb{R}^3$, sino valores escalares asociados al parámetro transversal. La superficie se construye directamente a nivel de puntos de control, aplicando el escalamiento únicamente a las coordenadas longitudinales de la curva generatriz, lo que permite generar superficies cilíndricas NURBS con variación de sección controlada de forma local.

```
class Nurbs_surface_escalar(Nurbs_surface_base):
    def __init__(self, pts1, pts2, pts3, *args):
        super().__init__(pts1, pts2, *args)
        self.pts3 = np.array(pts3, dtype=float)
    def P_ij(self, i, j):
        return [
            self.pts3[j] * self.pts1[i][0],
            self.pts2[j][1],
            self.pts3[j] * self.pts1[i][2] +
            self.pts2[j][2]
        ]
```

En el siguiente ejemplo se muestra una circunferencia que, durante su traslación, experimenta una contracción progresiva hasta colapsar en un punto y posteriormente recupera su tamaño original.

Superficie traslacional con escalar
Paso 1: Definimos todo lo necesario

```
# Construimos los datos
C_gen = [[2,0,1], [2,0,2.7321], [0.5,0,1.8660], [-1,0,1],
         [0.5,0,0.1340], [2,0,-0.7321], [2,0,1]]
peso1 = [1, 1/2, 1, 1/2, 1, 1/2, 1]
nodo1 = [0, 0, 0, 1, 1, 2, 2, 3, 3, 3]
C_dir = [[0,0,0], [0,1.3879,0], [0,3.2705,0],
[0,5.0015,0], [0,7.3201,0]]
peso2 = [1, 1, 1, 1, 1]
nodo2 = [0, 0, 0, 1/2, 1/2, 1, 1, 1]
C_escalar = [1, 0.3773, 0.3, 0.3773, 1]
```

```
# Inicializamos el objeto
surf = Nurbs_surface_escalar(C_gen, C_dir, C_escalar,
peso1, peso2, nodo1, nodo2, 2, 2)
# Luego visualizamos
Sweep4 = Nurbs_superficie3D_graph(surf)
Sweep4.grafica(pdir=False, pgen=False, c_pdir=True,
c_pgen=True, enmallar=True, pts_ctrl=True,
superficie=True)
```

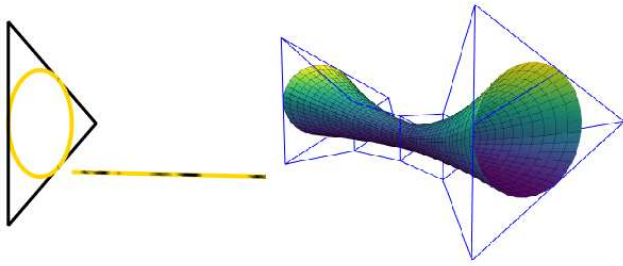


Fig 5. Superficie NURBS traslacional con contracción y expansión controlada mediante un escalar.

IV. COMPARACIÓN CON LIBRERÍAS EXISTENTES

Se realizó una comparación directa con superficies generadas mediante la librería geomdl, ampliamente utilizada para la construcción y manipulación de entidades NURBS en entornos computacionales. Para garantizar una comparación consistente, se emplearon las mismas curvas generatrices y trayectorias en ambos enfoques. La evaluación se llevó a cabo mediante el cálculo del error cuadrático medio (RMSE) entre las superficies generadas, considerando una discretización uniforme en el dominio paramétrico.

A. Superficie de barrido

En este caso correspondiente a una superficie de barrido generada a partir de una curva cerrada y una trayectoria no lineal, se obtuvo una discrepancia relativa del orden de:

$$RMSE = 6.06 \times 10^{-10}$$

Este resultado indica una alta concordancia entre ambas representaciones, lo que sugiere que el método propuesto reproduce adecuadamente la geometría obtenida mediante formulaciones tradicionales implementadas en geomdl.

B. Superficie con escalar

En este segundo caso se evaluó una superficie cilíndrica con factor escalar variable, donde la sección transversal experimenta contracciones y expansiones a lo largo de la trayectoria. El error obtenido fue:

$$RMSE = 5.20 \times 10^{-10}$$

La baja magnitud del error confirma que el enfoque basado en puntos de control permite modelar transformaciones más complejas sin pérdida significativa de precisión geométrica.

C. Análisis de tiempo de ejecución y uso de memoria

Además de la comparación geométrica, se realizó una evaluación del desempeño computacional entre el framework propuesto y la librería geomdl, considerando como caso de estudio la superficie cuatro. Para ello, se ejecutaron 100 repeticiones de cada método bajo las mismas condiciones.

Resultados con geomdl

- Tiempo total (100 ejecuciones): 1.161331 s
- Tiempo promedio por ejecución: 0.011613 s
- Uso actual de memoria: 8.57 KB
- Pico de memoria durante ejecución: 323.46 KB

Resultados con la implementación propuesta

- Tiempo total (100 ejecuciones): 1.083193 s
- Tiempo promedio por ejecución: 0.010832 s
- Uso actual de memoria: 8.49 KB
- Pico de memoria durante ejecución: 323.39 KB

Estos resultados indican que el enfoque basado en puntos de control mantiene un desempeño competitivo, con una leve ventaja en tiempo

V. RELACIÓN CON FORMULACIONES PARAMÉTRICAS ANALÍTICAS

El enfoque propuesto se basa conceptualmente en formulaciones clásicas de superficies (regladas, cilíndricas y de barrido), pero traslada su construcción al dominio discreto de los puntos de control NURBS. En este trabajo, las expresiones paramétricas no se utilizan como mecanismo de cálculo, sino como referencia interpretativa, mientras que la generación de superficies se realiza directamente mediante operaciones sobre puntos de control, pesos y vectores de nudos.

No obstante, la equivalencia formal entre la formulación continua y la construcción discreta no se desarrolla completamente en esta etapa. El presente trabajo prioriza la coherencia computacional del método, dejando como línea futura la formalización teórica rigurosa de esta correspondencia.

VI. DISCUSIONES

El framework propuesto demuestra que es posible reformular la construcción de superficies regladas, cilíndricas y de barrido sin recurrir a expresiones paramétricas analíticas explícitas, trasladando el proceso constructivo directamente al dominio de los puntos de control NURBS. Esta decisión metodológica reduce la brecha existente entre la formulación teórica clásica de superficies y su implementación computacional práctica, donde el diseño geométrico se realiza principalmente mediante la manipulación de estructuras discretas.

A diferencia de los enfoques tradicionales de superficies de barrido basados en marcos móviles, como los rotation minimizing frames, el método presentado evita el cálculo explícito de referencias diferenciales asociadas a la trayectoria.

Si bien estos marcos son eficaces para controlar la orientación del perfil y garantizar suavidad geométrica, introducen complejidad adicional y dependen de aproximaciones geométricas o del cálculo diferencial de la curva directriz. En contraste, el enfoque basado en puntos de control permite construir superficies coherentes desde una perspectiva puramente algebraica y computacional, lo que simplifica la implementación y favorece su integración en entornos de software orientados al diseño interactivo.

Asimismo, en comparación con trabajos enfocados en manufactura asistida por computadora, donde las superficies de revolución y barrido se utilizan principalmente como soporte para la generación de trayectorias de mecanizado, el presente trabajo no persigue objetivos de optimización industrial ni de control numérico. En su lugar, se centra en el modelado geométrico computacional, proporcionando una estructura unificada y extensible para la generación de superficies NURBS que puede ser posteriormente utilizada tanto en contextos de visualización como de diseño geométrico asistido por computadora.

Un aspecto relevante del framework es la introducción de superficies cilíndricas con factor escalar dependiente del parámetro transversal, lo que permite modelar contracciones y expansiones progresivas de la sección generatriz durante la traslación. Esta extensión amplía las capacidades de las superficies cilíndricas clásicas y muestra que el enfoque basado en puntos de control facilita la incorporación de transformaciones adicionales sin alterar la estructura general del modelo. La variación de sección se logra de manera local y controlada, reforzando una de las principales ventajas de las representaciones NURBS.

Desde el punto de vista computacional, la construcción directa de superficies a nivel de puntos de control permite generar mallados suaves y coherentes, manteniendo una correspondencia clara entre los datos de entrada y la geometría resultante. Esta propiedad resulta especialmente valiosa en aplicaciones donde el control local de la forma es prioritario, ya que modificaciones puntuales en los puntos de control se reflejan de manera predecible en la superficie final.

Si bien el trabajo se ha centrado en superficies regladas, cilíndricas y de barrido, la metodología propuesta es conceptualmente extensible a otras familias de superficies definidas por procesos constructivos similares, el framework presentado no debe interpretarse como un conjunto cerrado de técnicas, sino como una base computacional sobre la cual pueden desarrollarse modelos más generales, incorporando nuevas transformaciones geométricas o integrándose con sistemas CAD y herramientas de visualización científica más avanzadas.

VII. CONCLUSIONES

En este trabajo se presentó un framework computacional para la construcción de superficies regladas, cilíndricas y de barrido basado exclusivamente en puntos de control NURBS, eliminando la dependencia de formulaciones paramétricas

analíticas explícitas. Las expresiones paramétricas no se utilizan como mecanismo de cálculo, sino como referencia interpretativa, mientras que la generación de superficies se realiza directamente mediante operaciones sobre puntos de control, pesos y vectores de nudos. Este enfoque permite trasladar el proceso de construcción geométrica al dominio discreto, coherente con los flujos de trabajo utilizados en sistemas de diseño geométrico asistido por computadora.

La metodología propuesta demuestra que es posible generar superficies NURBS complejas mediante combinaciones afines y transformaciones controladas a nivel de puntos de control, manteniendo un control geométrico local preciso y una estructura computacional clara. La inclusión de superficies cilíndricas con factor escalar amplía las capacidades del modelo, permitiendo la variación progresiva de la sección transversal durante la traslación.

Los ejemplos computacionales presentados evidencian la flexibilidad y extensibilidad del framework, así como su adecuación para aplicaciones de modelado geométrico y visualización científica. Como trabajo futuro, se plantea la extensión de esta metodología a superficies más generales y su integración con entornos CAD y herramientas de diseño paramétrico avanzado.

AGRADECIMIENTO/RECONOCIMIENTO

Los autores expresan su agradecimiento a la Universidad Nacional de Piura por el apoyo institucional brindado durante el desarrollo de este trabajo.

VIII. REPOSITORIO

La implementación completa de las clases, los algoritmos de transformación de puntos de control y los scripts de visualización 3D se encuentran disponibles públicamente en el repositorio:

https://github.com/alexanwww/SurfaceOfSweep_NURBS.

REFERENCIAS

- [1] J. Cordero and J. Cortés, "Curvas y Superficies para Modelado Geométrico", México: Alfaomega, 2002.
- [2] D. Salomon, "Curves and Surfaces for Computer Graphics, Springer, 2006.
- [3] C. Lehmann, "Geometría Analítica", México: Editorial Limusa, 1989.
- [4] G. Farin, "Curves and Surfaces for CAGD (Computer Aided Graphics and Design)", San Diego: Academic Press, 2001.
- [5] L. Piegl and W. Tiller, The NURBS Book, 2nd ed. Berlin, Heidelberg: Springer, 1997, doi: 10.1007/978-3-642-59223-2.
- [6] W. Wang and B. Joe, "Robust computation of the rotation minimizing frame for sweep surface modeling," Computer-aided Design, vol. 29, no. 5, pp. 379–391, May 1997, doi: 10.1016/S0010-4485(96)00077-2.
- [7] S. Omirou, "NC machining for revolved and swept surfaces with free-form profiles," Journal of Materials Processing Technology, vol. 132, no. 1–3, pp. 332–339, Jan. 2003, doi: 10.1016/S0924-0136(02)00944-5.
- [8] R. T. U. Guzman, Y. A. C. Maza, and R. P. S. Silupu, "Generation and Visualization of NURBS Surfaces Derived from Controlled NURBS Curves," in Computational Science and Its Applications – ICCSA 2025, vol. 15648, O. Gervasi et al., Eds. Cham, Switzerland: Springer, 2025, pp. [páginas si las tienes], doi: 10.1007/978-3-031-97000-9_1.

- [9] R. T. Urbina Guzmán, Y. A. Castillo Maza, R. P. Santamaria Silupu, J. E. Jaramillo Atoche, M. H. Garcia Saba, V. H. Silupu Ortega, y J. L. Torres Lizana, "Construction of Surfaces Defined from Planar Curves in Python (#1272)," en *Engineering, Artificial Intelligence, and Sustainable Technologies in service of society*, México, jul. 16–18, 2025.
- [10] Y. A. Castillo Maza, R. P. Santamaria Silupu, R. T. Urbina Guzmán, J. K. Jiménez Vilcherrez, F. M. Velasquez Fernández, H. G. Sosa Calle, y H. A. Sandoval Calderón, "Coons Surfaces Expressed as B-Spline Surfaces in Python (#1963)," en *Engineering, Artificial Intelligence, and Sustainable Technologies in service of society*, México, jul. 16–18, 2025.
- [11] L. Cuervo, N. Cuervo, and J. Cuervo, "Iniciando a programar con Python: Guía básica de programación". Editorial de la Universidad Pedagógica y Tecnológica de Colombia - UPTC, 2024.
- [12] N. Miolane et al., "Introduction to Geometric Learning in Python with Geomstats," pp. 48–57, Jul. 2020, doi: 10.25080/MAJORA-342D178E007.
- [13] Y. A. Castillo-Maza *et al.*, "Modeling of Spheres and Tori of Revolution Using NURBS Surfaces in Python," en *Advanced Research in Technologies, Information, Innovation and Sustainability (ARTIIS 2025)*, T. Guarda, F. Portela, M. F. Augusto, y J. R. Coronado-Hernández, Eds. Cham, Suiza: Springer, 2026, pp. [números de página si los tienes], doi: 10.1007/978-3-032-16761-3_2.