

# SỬ DỤNG NGÔN NGỮ LẬP TRÌNH VISUAL BASIC ĐỂ XÂY DỰNG MẶT TỰ DO TRONG AUTOCAD

THS.KS PHÙNG QUANG MINH | TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

## Tóm tắt

Việc thiết kế kiến trúc muốn đạt kết quả tốt hơn thì việc lựa chọn hình dáng kiến trúc công trình rất quan trọng, ngày nay bề mặt công trình được sử dụng nhiều trong thiết kế là các bề mặt phức tạp, tùy biến (mặt tự do). Bài báo nghiên cứu phương pháp xây dựng mặt trong AutoCad và đưa ra phương pháp xây dựng mặt tự do là dùng ngôn ngữ lập trình Visual Basic để viết mã code chương trình con tạo mặt tự do và được nhúng vào phần mềm AutoCad. Đồng thời, bài báo cũng đưa ra thuật toán xây dựng mặt tự do với đường chuẩn là đường xoắn ốc trụ và đường sinh thay đổi trong quá trình tạo mặt để máy tính điện tử tự động tạo mặt, cũng như tạo cơ sở dữ liệu có thể chuyển đổi sang các phần mềm ứng dụng khác trong thiết kế. Phương pháp này giúp người làm thiết kế tạo mặt tự do một cách tự động, giảm thời gian thiết kế, và hiện thực hóa các ý tưởng thiết kế đơn giản hơn.

## Abstract

In order to achieve better results in architectural design, choosing the shape of the building is very important. Nowadays, the building surfaces used a lot in design are complicated, customized ones (free surfaces). The scientific article analysis the method of creating free surface models in AutoCad and proposes a method of creating free surface models, which use the Visual Basic programming language to write a subprogram code in order to create free surface models and embed it in AutoCad software. Simultaneously, the article also provides an algorithm to build a free surface with a calibration curve that is a cylindrical spiral and a surface lines which changes during the process of creating a surface so that an electronic computer can automatically create a surface, as well as create a database that can convert to other application design software. This method will help designers in creating free surfaces automatically, as well as reducing designing time, and actualizing simpler design ideas.

## ĐẶT VẤN ĐỀ

Những thập kỷ trước, ngành công nghiệp kiến trúc xây dựng đã trải qua những thay đổi lớn với việc phát minh ra "phần mềm hỗ trợ công tác thiết kế" (Computer Aided Design - CAD). Từ đó, càng nhiều những phần mềm thiết kế kiến trúc khác của các hãng như Autodesk, Bentley, Gehry Technologies, Graphisoft lần lượt ra đời. Ngành kiến trúc xây dựng vốn chỉ quen vẽ bằng tay nay đã làm quen với việc sử dụng máy tính như một công cụ để hỗ trợ cho việc thiết kế.

Tuy nhiên, công việc thiết kế kiến trúc, quy hoạch muốn thực hiện tốt thì người kiến trúc sư phải có các kiến thức chuyên sâu trong nhiều lĩnh vực khác nhau. Và đặc biệt là việc thiết kế các hình khối kiến trúc, trong đó bề mặt kiến trúc là một trong những yếu tố quan trọng quyết định đến hình dáng kiến trúc của công trình. Mặt khác, hiện nay với thời đại 4.0, việc sử dụng các dạng mặt tự do được áp dụng phổ biến hơn, cùng với sự hỗ trợ của máy tính điện tử (máy tính) thì việc xây dựng và sử dụng các mặt phức tạp, tùy biến (mặt tự do) ngày càng xuất hiện phổ biến.

Từ các lý do trên nên việc nghiên cứu phương pháp dùng ngôn ngữ lập trình Visual Basic để xây dựng mặt tự do trong phần mềm AutoCad là cần thiết và có thể áp dụng trong thực tế.

## NỘI DUNG NGHIÊN CỨU

### Cơ sở khoa học xây dựng mặt trong Autocad

Trong phần này bài báo đưa ra định nghĩa về mặt nói chung hay là mặt tự do và nguyên tắc tạo mặt.

#### Khái niệm về mặt tự do

Mặt là quỹ tích các vị trí một đường cong (gọi là đường sinh) chuyển động luôn tựa trên một đường cong khác (gọi là đường chuẩn). Như vậy, mặt tự do có thể là các mặt hình học đơn giản hoặc là các mặt cong phức tạp trong tự nhiên.

Các trường hợp cụ thể cần có thêm một số điều kiện nữa để có được mặt tự do cụ thể, đường sinh có thể là đường cong và đường cong này cũng có thể thay đổi thuộc tính theo quy luật trong quá trình chuyển động. Đường chuẩn có thể là đường cong cố định trước hoặc đường cong này cũng có thể thay đổi theo quy luật trong quá trình đường sinh chuyển động. Các quy luật thay đổi của đường sinh hay đường chuẩn tùy thuộc vào người tạo mặt.

#### Nguyên tắc tạo mặt trong Autocad

Một số mặt như mặt phẳng, mặt nón, mặt trụ, mặt cầu có thể được miêu tả bằng các phương trình toán học. Tuy nhiên, hầu hết các mặt tự do trong thực tế không thể biểu

diễn được bằng một biểu thức toán học. Người ta tìm cách biểu diễn gần đúng một mặt tự do bằng cách tách nó ra thành vô số các mảnh nhỏ phẳng hoặc cong tạo thành một lưới các mảnh mặt đa giác trong không gian. Việc đảm bảo cho mặt gần đúng này thoả mãn đi qua các điểm hoặc các đường biên đã cho với độ cong khống chế trước là đặc biệt quan trọng.

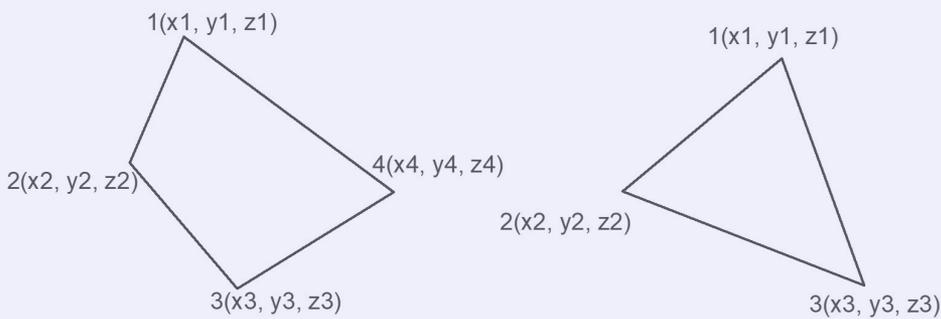
Trong thực tế sử dụng thì có nhiều cách tạo mặt khác nhau trong AutoCad, tuy nhiên bài báo đề cập đến cách tạo mặt trực tiếp từ các mảnh mặt như sau:

### Bước 1: Tạo các phần tử là các mảnh mặt 3D Face

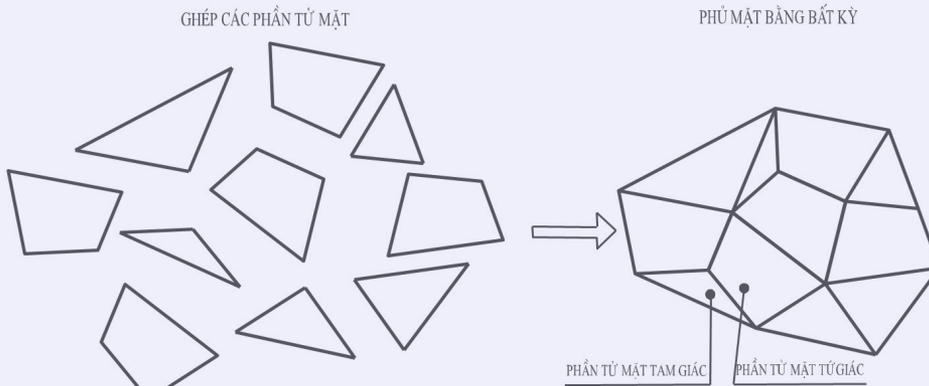
3D Face: Được thiết lập bằng 3 điểm hoặc 4 điểm bất kỳ có thể là đồng phẳng hoặc không đồng phẳng.

Trong hình H.1 thấy rằng đây là hai dạng mảnh mặt đơn giản nhất và nó là các phần tử cấu tạo nên các mặt phức tạp hơn. Các điểm này được đánh số thứ tự theo chiều ngược chiều kim đồng hồ.

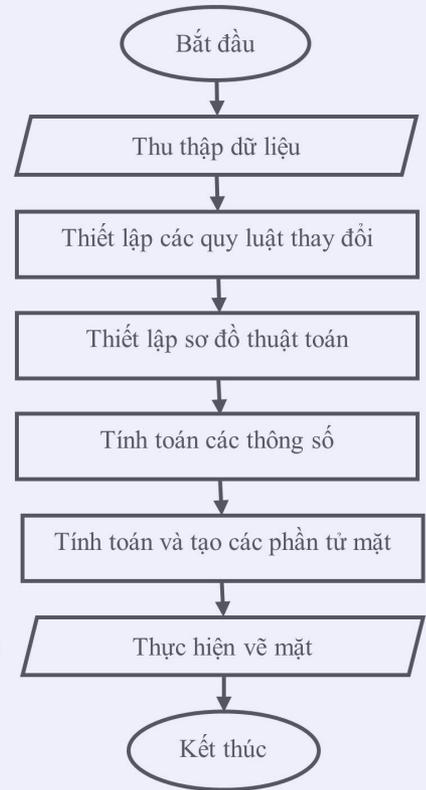
Trong phần tử mặt thiết lập bằng 4 điểm thì trong phần mềm Autocad lấy (1,2,4) và (2,3,4) là các mảnh mặt tam giác đồng phẳng, tuy nhiên, giao của hai mảnh mặt này là đoạn thẳng 24 chỉ là giả định vì thực tế là không có phần giao tuyến này và mặt tạo bởi (1,2,3,4) là mặt gẫy.



Hình H.1



Hình H.2



Hình H.3

**Bước 2: Tạo mặt từ hệ thống lưới phần tử mặt**

Lưới mặt: Được thiết lập bằng hệ thống lưới điểm. Lưới điểm này có thể là ô lưới 3 điểm hay ô lưới 4 điểm. Cứ mỗi ô lưới là một phần tử mặt.

Trên hình H.2 thấy rằng mặt này được tạo ra bởi 10 mảnh mặt phần tử, mỗi một mảnh mặt được thiết lập trên 3, 4 điểm quản lý tại 3, 4 nút. Hay nói cách khác mặt này được tạo ra khi ghép 10 mảnh mặt phần tử.

Về nguyên tắc bằng cách này có thể tạo được các mặt phức tạp hơn, chúng ta chỉ cần: Chia mặt cần dựng theo một hệ thống lưới điểm (ô lưới 3 điểm hay 4 điểm), xác định các đỉnh quản lý này thuộc mặt cần dựng; Dùng cách tạo mặt 3DFace để tạo các mảnh mặt phần tử, ghép chúng lại theo lưới điểm chia ở trên sẽ được mặt cần dựng.

Như vậy, với một mặt tự do nào đó thì việc xây dựng mặt trong AutoCad là hoàn toàn có thể thực hiện dựa vào phương pháp như trên. Tùy vào yêu cầu của người sử dụng để có thể đưa ra cách chia, ghép các mảnh mặt với độ chính xác theo yêu cầu của người sử dụng.

*Quy trình tạo mặt tự do trong Autocad*

Hình H.3 thể hiện sơ đồ thuật toán quy trình xây dựng mặt tự do trong AutoCad.

Trong việc thu thập số liệu cần xác định được các thông số về độ chính xác, các quy luật thay đổi của đường chuẩn và đường sinh.

Trong bước thiết lập các quy luật thay đổi cần thành lập các phương trình toán học mô tả các quy luật thay đổi này. Các phương trình này thông thường thể hiện dưới dạng phương trình tham số để người sử dụng có thể thay đổi các quy luật theo ý muốn.

Việc thiết lập sơ đồ thuật toán tạo mặt là rất quan trọng trong việc tạo mặt. Sau khi có sơ đồ thuật toán thì người dùng có thể dùng bất kỳ ngôn ngữ lập trình nào để vẽ mặt trong AutoCad, tuy nhiên bài báo sẽ dùng ngôn ngữ lập trình Visual Basic để vẽ mặt.

Việc tính toán các thông số là số hóa các tọa độ của các điểm thuộc các mảnh mặt trong mặt tự do. Các tọa độ này phụ thuộc vào quy luật thay đổi, độ chính xác yêu cầu...

Tính toán và tạo các phần tử mặt là dùng hệ lưới điểm ở bước trên để phủ mặt 3DFace lên hệ lưới điểm này, độ mịn của mặt phụ thuộc vào việc tính toán các điểm lưới của mặt.

**Xây dựng mặt tự do trong Autocad**

Trong phần này, bài viết sẽ xây dựng mặt tự do với đường chuẩn là một đường xoắn ốc trụ, và đường sinh là đường tròn với quy luật thay đổi đường sinh là tăng đường kính

trong quá trình chuyển động.

*Thiết lập các quy luật thay đổi*

**Với đường chuẩn là đường xoắn ốc trụ**

Đường xoắn ốc trụ là quỹ tích những vị trí của một điểm chuyển động đều trên một đường thẳng l trong khi l chuyển động đều để tạo thành một mặt trụ. Với phương trình tham số như sau.

Đường xoắn ốc trụ là đường xoắn ốc trên mặt trụ (hướng đường sinh t và đường chuẩn (c) là vòng tròn đáy trụ), bước của đường xoắn là b.

Trong phần này xây dựng đường xoắn ốc trụ trên mặt giá mang có đường chuẩn (c) (đường tròn) và hướng đường sinh là thẳng đứng (theo trục z trong hệ tọa độ Đécác).

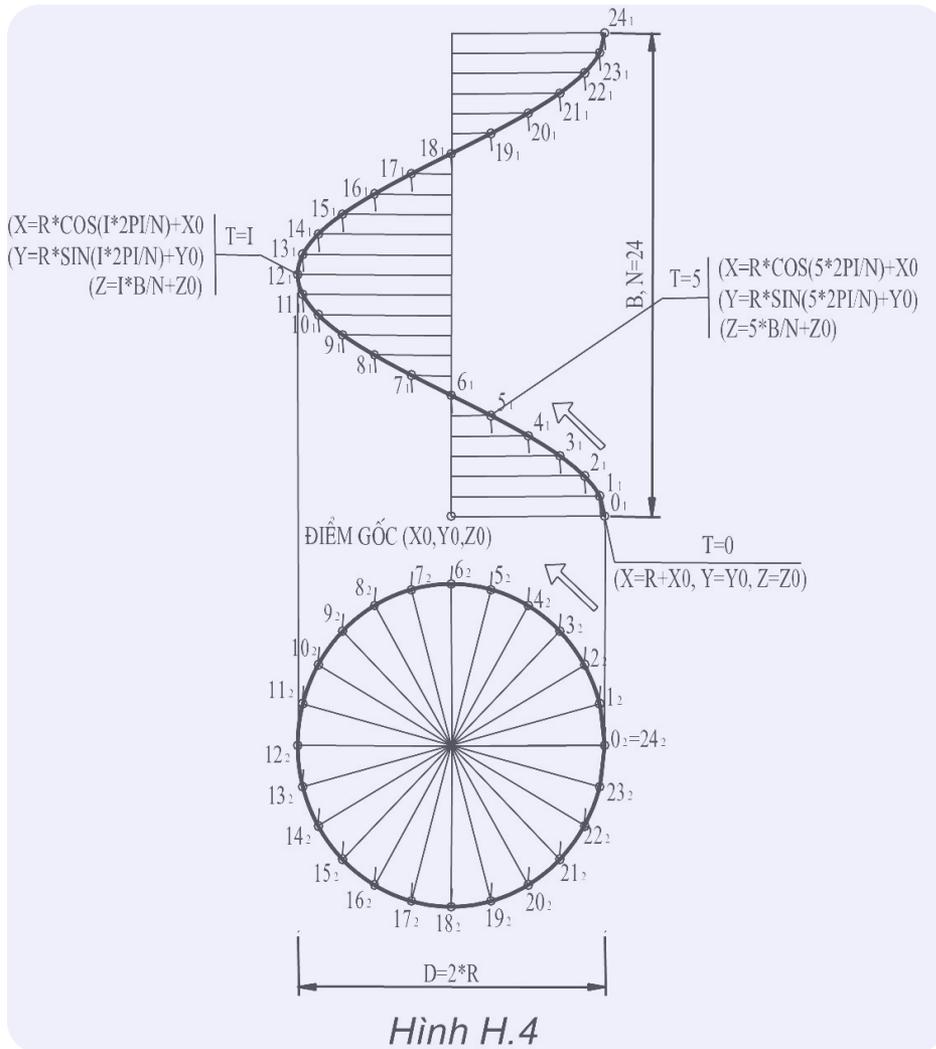
Điểm gốc O(x0, y0, z0) là trục của đường xoắn ốc trụ. Vậy điểm thuộc đường xoắn ốc trụ và thuộc tia hợp với trục x một góc là t sẽ có tọa độ:

$$z = b * t / n + z_0$$

$$x = R * \text{Cos}(t*2*pi/n) + x_0 \quad (1)$$

$$y = R * \text{Sin}(t*2*pi/n) + y_0$$

Phương trình tham số trên là phương trình tham số của đường xoắn ốc trụ, tham số t.



Hình H.4

Với đường sinh là đường tròn

Với mặt vỏ ốc có đường sinh tròn thì việc xây dựng phương trình cụ thể bằng cách xây dựng phương trình đường tròn tâm M, bán kính r = MN/2 trong mặt phẳng Oxz và dịch hệ trục tọa độ về gốc tọa độ Oxyz.

Phương trình đường tròn: cho chúng ta tọa độ điểm thuộc đường tròn (x,y,z).  
 Phương trình của mặt:

$$[x \ y \ z] = [x' \ y' \ z'] * T * R_z * R_y$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_M & -y_M & -z_M & 1 \end{bmatrix} R_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_y = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Lập sơ đồ thuật toán

Với đường chuẩn là đường xoắn ốc trụ, sơ đồ 1

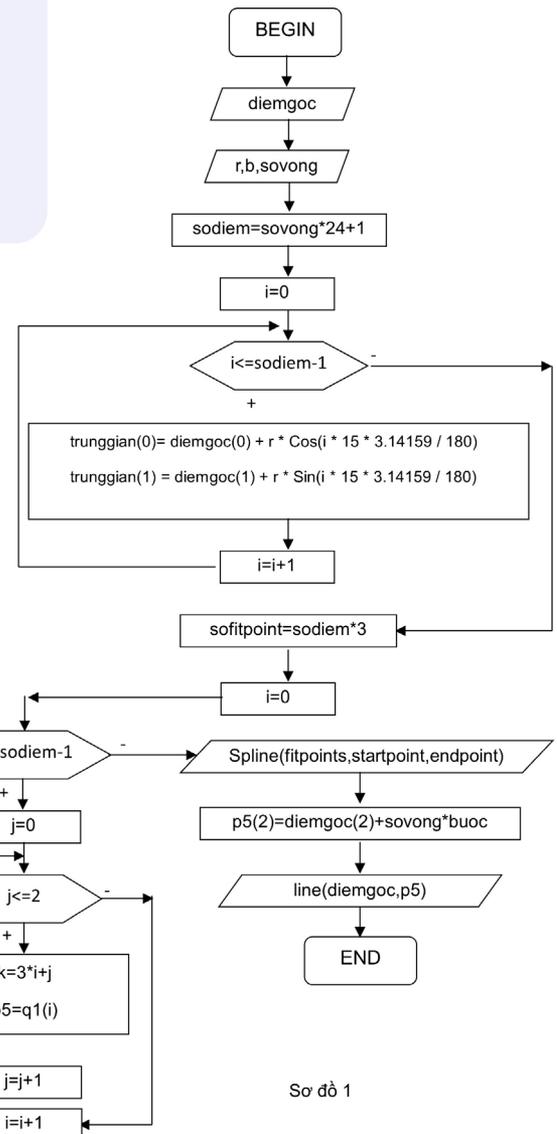
- o Bắt đầu;
- o Nhập vào điểm gốc (là điểm chân của trục đường xoắn ốc);
- o Nhập vào:

R: bán kính của mặt trụ tròn xoay  
 b: độ lớn của bước của đường xoắn ốc trụ  
 sovong: tổng số vòng xoắn  
 Số điểm chia trên một bước (trong bài báo chọn n=24);

- o Gán sodiem=sovong\*n+1 (trên một vòng xoắn được chia ra làm 24 khoảng bằng 25 điểm chia);
- o Gán số đếm i=0;
- o Tính toán tọa độ các điểm trên đường xoắn ốc theo phương trình (1) (các điểm quản lý này chính là các điểm chia trên đường xoắn ốc cần vẽ)

- $z = b * i / n + z0$
- $x = R * \cos(i*2pi/n) + x0$   
( $i = [0 \div \text{sodiem}]$ )
- $y = R * \sin(i*2pi/n) + y0$

o Gán tọa độ các điểm này cho một mảng quản lý điểm q(i);  
 o Gán các tọa độ xyz của mảng điểm q(i) sang mảng fitpoint(k), mảng fitpoint(k) này sẽ lưu các giá trị về tọa độ của đường xoắn ốc trụ và mảng này sử dụng được để vẽ đường spline có các điểm quản lý là các điểm thuộc mảng điểm (q),



Sơ đồ 1

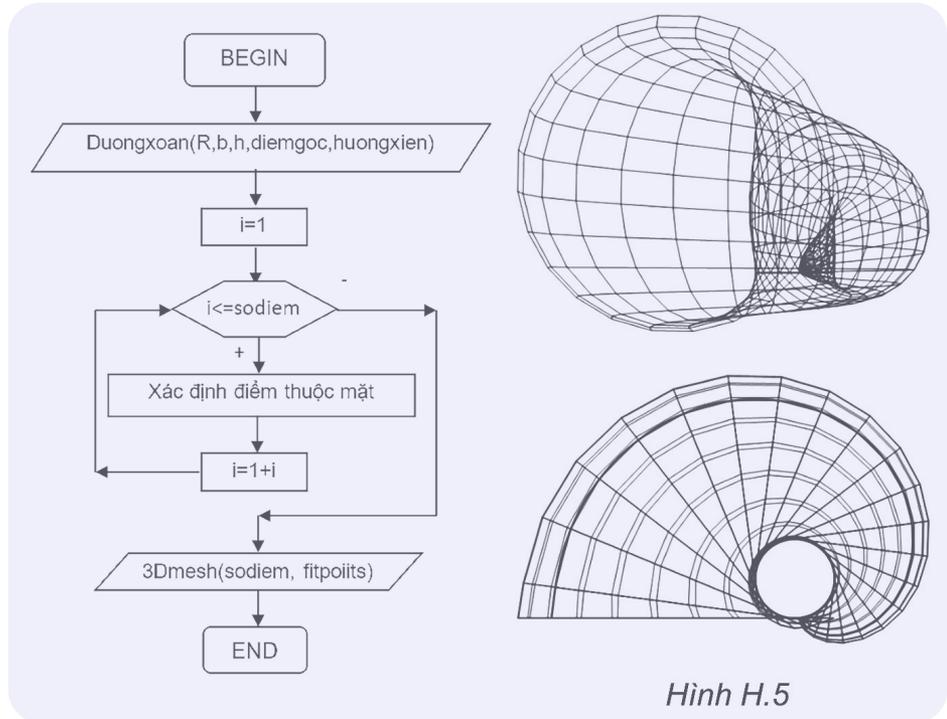
i	Mảng điểm	fitpoint (chuỗi số dữ liệu)
i	$\begin{cases} x = R\cos(i2\pi / n) + x_0 \\ y = R\sin(i2\pi / n) + y_0 \\ z = ib / n + z_0 \end{cases}$	$\text{fitpoint int} \begin{cases} \text{fitpoint}(3) = R\cos(i2\pi / n) + x_0 \\ \text{fitpoint}(4) = R\sin(i2\pi / n) + y_0 \\ \text{fitpoint}(5) = ib / n + z_0 \end{cases}$

Bảng 1:

- o Tính toán điểm trên của trục đường xoắn ốc trụ điểm p5;
- o Vẽ đường xoắn ốc trụ bằng đường spline;
- o Vẽ đoạn thẳng là trục của đường xoắn ốc trụ, đoạn thẳng này có điểm đầu là diemgoc và điểm cuối là p5;
- o Kết thúc.

**Với đường sinh là đường tròn, sơ đồ 2, Hình H.5**

- o Bắt đầu;
- o Nhập vào đường xoắn ốc trụ (xem sơ đồ 1);
- o Tính toán các tọa độ của các điểm thuộc lưới mặt tự do theo (2);
- o Gán các giá trị tọa độ các điểm thuộc lưới cho mảng điểm quản lý a(i);
- o Gán các tọa độ x, y, z cho mảng dữ liệu point của lưới mặt;
- o Vẽ mặt;
- o Kết thúc.



Hình H.5

**Viết code bằng ngôn ngữ lập trình Visual Basic**

**Tạo đường chuẩn hình H.6; H.7**

- o Tạo biểu tượng chương trình con bằng Visual Basic
- o Viết code cho đường chuẩn Private Sub cmdDiemgoc\_Click()

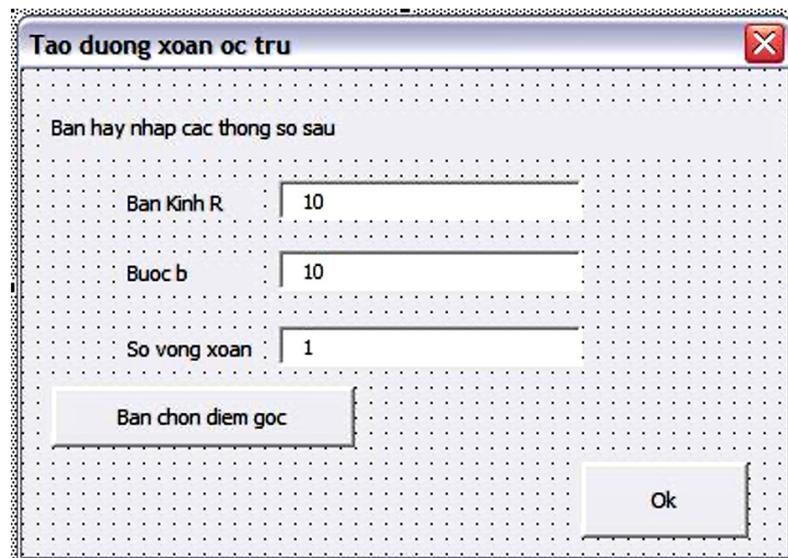


**- 'Xác định điểm gốc là chân trục của đường xoắn ốc trụ .**

```
UserForm.Hide
MsgBox ("Ban chọn điểm gốc")
diemgoc = ThisDrawing.Utility.GetPoint(
NL & "Điểm gốc")
trunggian = diemgoc
UserForm.Show
End Sub
Private Sub cmdok_Click()
UserForm3.Hide
```

**- 'Phần khai báo biến địa phương trong chương trình**

```
Dim bankinh, L, buoc As Double
Dim sovong As Integer
Dim q1() As Variant
Dim p5 As Variant
```



Hình H.6

```
Dim sodiem As Integer
Dim i As Integer
Dim tyle As Double
Dim lineObj As AcadLine
```

**'Phần gán các giá trị trong các ô nhập giá trị**

```
buc = Textb.Text
sovong = Textth.Text
r = TextR.Text
sodiem = sovong * 24 + 1 'Một bước được
chia ra làm 24 phần bằng nhau.
ReDim q1(0 To sodiem - 1) As Variant
```

**'Phần tính toán các giá trị của các điểm thuộc đường xoắn ốc trụ.**

```
trunggian = diemgoc
i = 0
For i = 0 To (sodiem - 1) Step 1
    trunggian(0) = diemgoc(0) + r * Cos(i * 15 *
    3.14159 / 180)
    trunggian(1) = diemgoc(1) + r * Sin(i * 15 *
    3.14159 / 180)
    trunggian(2) = diemgoc(2) + buc * i / 24
    q1(i) = trunggian
Next i
MsgBox (Str(i))
```

**' Tạo đường Spline là đường xoắn ốc trụ**

```
Dim splineObj As AcadSpline
Dim startTan(0 To 2) As Double
Dim endTan(0 To 2) As Double
Dim fitPoints() As Double
Dim a() As Double
Dim p1, p2, p3, p4 As Variant
Dim softpoint, d, k, j As Integer
```

**' Định dạng cho biến là tiếp tuyến điểm đầu và cuối**

```
startTan(0) = 0.5: startTan(1) = 0.5:
startTan(2) = 0
endTan(0) = 0.5: endTan(1) = 0.5:
endTan(2) = 0
```

**' Định dạng cho biến trung gian**

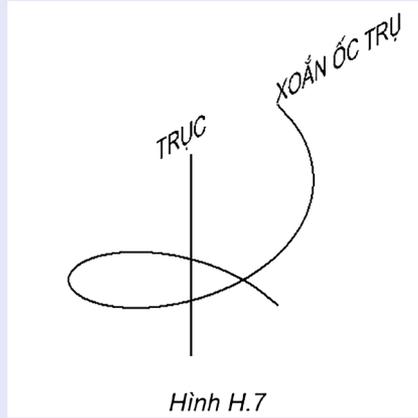
```
p1 = q1(0)
p2 = q1(1)
p3 = q1(sodiem - 2)
p4 = q1(sodiem - 1)
```

**' Tính toán các tiếp tuyến đầu và cuối**

```
startTan(0) = p2(0) - p1(0): startTan(1) =
p2(1) - p1(1): startTan(2) = p2(2) - p1(2)
endTan(0) = p4(0) - p3(0): endTan(1) = p4(1) -
p3(1): endTan(2) = p4(2) - p3(2)
```

**' Chuyển mảng điểm thành chuỗi dãy số fitpoint**

```
softpoint = sodiem * 3
MsgBox ("Tong so fitpoints la: " +
Str(softpoint))
ReDim a(0 To softpoint - 1) As Double
```



Hình H.7

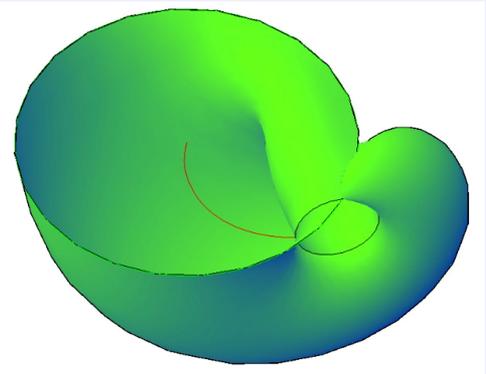
```
For i = 0 To (sodiem - 1) Step 1
    For j = 0 To 2 Step 1
        k = 3 * (i) + j
        p5 = q1(i)
        a(k) = p5(j)
    Next j
Next i
MsgBox (Str(k))
ReDim fitPoints(0 To softpoint - 1)
As Double
For i = 0 To (softpoint - 1) Step 1
    fitPoints(i) = a(i)
Next i
Set splineObj = ThisDrawing.ModelSpace.
AddSpline(fitPoints, startTan, endTan)
' Vẽ trục của đường xoắn ốc trụ
p5 = diemgoc
p5(2) = diemgoc(2) + sovong * buc
Set lineObj = ThisDrawing.Model-
Space.AddLine(diemgoc, p5)
ZoomAll
ThisDrawing.Application.Update
UserForm.Show
End
End Sub
```

**Tạo đường sinh và tạo mặt hình H.8**

```
Private Sub CommandButton6_Click()
UserForm1.Hide
Dim sset As AcadSelectionSet 'Define
sset as a SelectionSet object
```

**'Set sset to a new selection set named SS1 (the name doesn't matter here)**

```
Dim p() As Variant
Dim r() As Variant
Dim line As AcadLine
Dim p1, p5, p2, p3, p4, p6, p7, p8 As Variant
Dim t1, t2 As Variant
Dim q As Integer
Dim t As Double
Dim splineObj As AcadSpline
Dim startTan(0 To 2) As Double
```



Hình H.8

```
Dim endTan(0 To 2) As Double
Dim fitPoints() As Double
Dim a() As Double
Dim NL As String
NL = Chr(13) & Chr(10)
Dim dem, c, d, i, j, k, h, L As Integer
```

**'Lay toa do fitpoints va ve cac duong sinh của mặt**

**'p5 = ThisDrawing.Utility.GetPoint(, NL & "Diem mat")**

```
Set sset = ThisDrawing.SelectionSets.
Add("SS1")
sset.SelectOnScreen
```

**'Prompt user to select objects**

```
Dim ent As Object
```

**Define ent as an object**

```
For Each ent In sset
```

**'Loop through the SelectionSet collection**

```
With ent
    If StrComp(ObjectName, "AcDb-
spline", 1) = 0 Then
        q = ent.NumberOfFitPoints
        ReDim r(0 To q - 1) As Variant
        ReDim p(0 To q - 1) As Variant
```

```
p3 = ent.GetFitPoint(0)
p4 = ent.GetFitPoint(1)
p2 = ent.GetFitPoint(q - 1)
t1 = p4
t1(0) = p4(0) - p3(0)
t1(1) = p4(1) - p3(1)
t1(2) = p4(2) - p3(2)
p4 = ent.GetFitPoint(q - 2)
t2 = p4
t2(0) = p2(0) - p4(0)
t2(1) = p2(1) - p4(1)
t2(2) = p2(2) - p4(2)
p4 = p3
p4(0) = t2(0) * (p3(2) - p2(2)) / t2(2) + p2(0)
p4(1) = t2(1) * (p3(2) - p2(2)) / t2(2) + p2(1)
p4(2) = p3(2)
r(0) = p3
r(q - 1) = p4
p(0) = ent.GetFitPoint(0)
```

```

p(q - 1) = ent.GetFitPoint(q - 1)
Set line = ThisDrawing.ModelSpace.
AddLine(r(q - 1), p(q - 1))
For i = 1 To q - 2 Step 1
    p(i) = ent.GetFitPoint(i)
    p1 = p(i)
    p6 = ent.GetFitPoint(i - 1)
    p7 = ent.GetFitPoint(i + 1)
    p8 = p(i)
    t = (p3(2) - p1(2)) / (p7(2) - p6(2))
    p8(0) = t * (p7(0) - p6(0)) + p1(0)
    p8(1) = t * (p7(1) - p6(1)) + p1(1)
    p8(2) = p3(2)
    r(i) = p8
    Set line = ThisDrawing.ModelSpace.
AddLine(r(i), p(i))
Next i
End If
End With
ent.color = acBlue 'Change the
Color property to blue
ent.Update 'Update the entity
on screen
Next ent
sset.Delete

```

### 'Ve duong spline chan

```

startTan(0) = 0.5: startTan(1) = 0.5: startTan(2) = 0
endTan(0) = 0.5: endTan(1) = 0.5: endTan(2) = 0
p1 = r(0)
p2 = r(1)
p3 = r(q - 2)
p4 = r(q - 1)
startTan(0) = p2(0) - p1(0): startTan(1) = p2(1) - p1(1): startTan(2) = p2(2) - p1(2)
endTan(0) = p4(0) - p3(0): endTan(1) = p4(1) - p3(1): endTan(2) = p4(2) - p3(2)
d = q * 3
MsgBox ("Tong so fitpoints la: " + Str(q))
ReDim a(0 To d - 1) As Double
For i = 0 To q - 1 Step 1
    For j = 0 To 2 Step 1
        k = 3 * i + j
        p5 = r(i)
        a(k) = p5(j)
    Next j
Next i
'MsgBox (Str(k))
ReDim fitPoints(0 To d - 1) As Double
For i = 0 To d - 1 Step 1
    fitPoints(i) = a(i)
Next i
Set splineObj = ThisDrawing.ModelSpace.
AddSpline(fitPoints, startTan, endTan)
ZoomAll
ThisDrawing.Application.Update

```

### 'Ve mat bang Mesh

```

Dim meshObj As AcadPolygonMesh
Dim mSize, nSize, count As Integer
Dim points() As Double
Dim luoidiem() As Variant
Dim duongkinh As Double
Dim a1, b1, c1 As Double

```

### 'Gan so luong points cho Mesh

```

ReDim luoidiem(0 To 24, 0 To (q - 1)) As Variant
For i = 0 To (q - 1) Step 1
    p7 = r(i)
    p8 = p(i)
    a1 = p8(0) - p7(0)
    b1 = p8(1) - p7(1)
    c1 = p8(2) - p7(2)
    duongkinh = Sqr(a1 * a1 + b1 * b1 + c1 * c1)
    'MsgBox (Str(duongkinh))
    For j = 0 To 24 Step 1
        If duongkinh = 0 Then
            luoidiem(j, i) = p7
        Else
            p5 = p7
            p5(0) = a1 * (Cos(15 * j * 3.14159 / 180) + c1 * Sin(15 * j * 3.14159 / 180)) / Sqr(a1 * a1 + b1 * b1) / 2 + p7(0) / 2 + p8(0) / 2
            p5(1) = b1 * (Cos(15 * j * 3.14159 / 180) + c1 * Sin(15 * j * 3.14159 / 180)) / Sqr(a1 * a1 + b1 * b1) / 2 + p7(1) / 2 + p8(1) / 2
            p5(2) = (Sin(15 * j * 3.14159 / 180) * Sqr(a1 * a1 + b1 * b1) - c1 * Cos(15 * j * 3.14159 / 180)) / 2 + (p8(2) + p7(2)) / 2 + c1 * (Cos(15 * j * 3.14159 / 180) + c1 * Sin(15 * j * 3.14159 / 180)) / Sqr(a1 * a1 + b1 * b1) + p7(2)
            luoidiem(j, i) = p5
        End If
    Next j
Next i
d = 25 * q * 3
MsgBox ("Tong so diem la: " + Str(d))
ReDim points(0 To d - 1) As Double
'Mang luoi diem
h = 0
For i = 0 To (q - 1) Step 1
    For j = 0 To 24 Step 1
        For k = 0 To 2 Step 1
            p5 = luoidiem(j, i)
            points(h) = p5(k)
            h = h + 1
        Next k
    Next j
Next i
MsgBox ("Tong so point la: " + Str(h))

mSize = q: nSize = 25

```

### 'creates a 3Dmesh in model space

```

Set meshObj = ThisDrawing.ModelSpace.
Add3DMesh(mSize, nSize, points)
'Change the viewing direction of the viewport to better see the cylinder

```

```

Dim NewDirection(0 To 2) As Double
NewDirection(0) = -1: NewDirection(1) = -1:
NewDirection(2) = 1
ThisDrawing.ActiveViewport.Direction =
NewDirection
ThisDrawing.ActiveViewport = ThisDrawing.ActiveViewport
ZoomAll
UserForm1.Show
End Sub

```

## KẾT LUẬN

Bài viết đã đáp ứng được mục đích ban đầu đặt ra là sử dụng ngôn ngữ lập trình Visual Basic để xây dựng mặt tự do trong AutoCad, đưa ra một cách tạo mặt tự do một cách tự động bằng cách nhúng thêm vào AutoCad một chương trình con tạo mặt.

Bài báo là một tài liệu hướng dẫn tạo mặt tự do trong AutoCad và từ dữ liệu tạo mặt có thể chuyển đổi cơ sở dữ liệu giữa các phần mềm khác nhau cũng như từ lý thuyết về tạo mặt sang mô hình trong máy tính được thể hiện trong phần mềm AutoCAD.

Bài viết mới chỉ đi nghiên cứu tạo mặt có đường chuẩn là đường xoắn ốc trụ, đường sinh là đường tròn thay đổi đường kính trong quá trình chuyển động để tạo mặt. Nên với các mặt tự do khác có thể làm tương tự/.

## TÀI LIỆU THAM KHẢO

1. PGS. TS. Đặng Văn Cứ (chủ biên), GVC. Nguyễn Quang Cự, PGS. TS. Đoàn Như Kim, PGS. Dương Tiến Thọ, *Giao tiếp đồ họa kỹ thuật Xây dựng, nhà xuất bản Giáo Dục, Hà Nội, 2006.*
2. Hoàng Văn Thân (chủ biên), Đoàn Như Kim, Dương Tiến Thọ, *Hình học họa hình, nhà xuất bản Khoa học và kỹ thuật, Hà Nội, 2005.*
3. В.О.Гордон, М.А.Семенов-Огиевский КУРС НАЧЕРТАТЕЛЬНОЙ, ГЕОМЕТРИИ, НЕГ Врф-ntkmcndj Dscifz irjlf 2000. (V.O. Gordon, M. A. Sementsov-Ogieyskiy *ÔN TẬP HÌNH HỌC HỌA HÌNH, Nhà xuất bản Đại học URE, năm 2000*)
4. Hoàng Văn Thân (chủ biên), Đoàn Như Kim, Dương Tiến Thọ, *Hình học họa hình, nhà xuất bản Khoa học và kỹ thuật, Hà Nội, 2005.*
5. Nguyễn Tiến, Ngô Quốc Việt, Phạm Nguyễn Tuấn Kỳ, *Kỹ thuật lập trình Visual Basic tự học trong 21 ngày, nhà xuất bản Giáo dục, 1997.*
6. Bài báo: *Tạp chí Kiến trúc Việt Nam, số 237-2021, Ths Phùng Quang Minh, Xây dựng mặt xoắn bằng máy tính điện tử để tiện áp dụng trong thiết kế và dạy học môn Hình học họa hình.*  
<https://kientrucvietnam.org.vn/paper/tap-chi-ktvn-so-237-2021/>