La Colonna Tortile

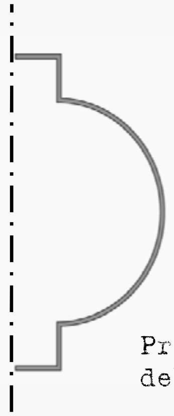
DEFINIZIONE:

Elemento architettonico verticale di sezione circolare composto di base, fusto e capitello. Sostegno a strutture sovrastanti, o con funzione meramente decorative e monumentale.

In particolare è detta TORTILE se presenta il fusto ritorto a forma di treccia, cioè attorcigliata in spire lungo l'asse verticale, tipica dello stile barocco.

L'Osservazione

L'osservazione dal vero delle decorazioni del portale principale, come di quelli secondari, ci ha potuto far notare come siano presenti diversi tipi di colonne tortili.

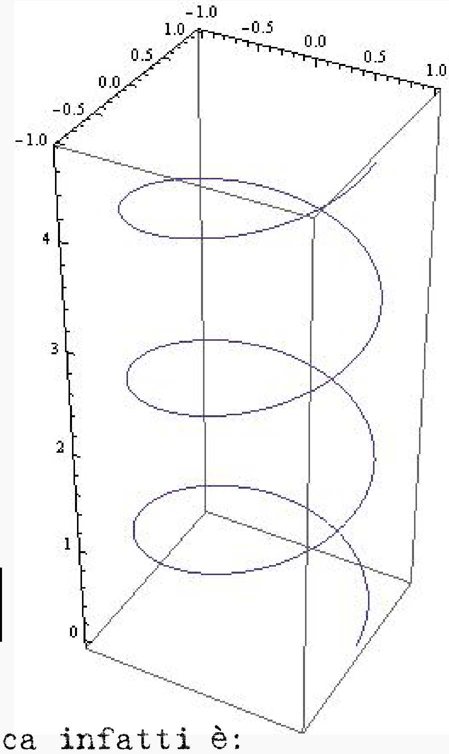


Tuttavia ho voluto analizzare l'unico elemento che interviene indistintamente su ciascuna tipologia. Tutte le colonne tortili decorative del duomo di Orvieto hanno infatti una "modanatura a toro", per così dire, rappresentata in sezione dalla figura riportata.

Profilo
della decorazione tortile

Esempi diversi di colonne tortili presenti nel Duomo di Orvieto

L'Elica

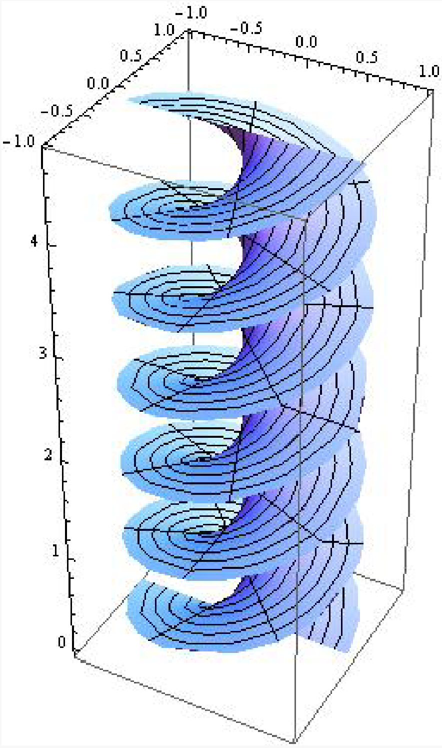


La curva che meglio rappresenta l'andamento di una generica colonna tortile è senza dubbio **ELICA CILINDRICA.**

DEFINIZIONE:

L'Elica cilindrica, definita da un asse,un raggio r e un passo p, è la traccia lasciata da un punto che si muove sulla superficie cilindrica di raggio r, ruotando attorno all'asse e avanzando a ogni giro di un passo p.

```
elica[a_,b_][t_]:={a*Cos[t], b*Sin[t],b*t}
```



L'eq. parametrica dell'Elica infatti è:

$$\begin{cases} x = a\cos(t) \\ y = b\,\mathrm{sen}(t) \\ z = bt \end{cases}$$

dove a è il raggio e b determina il passo.

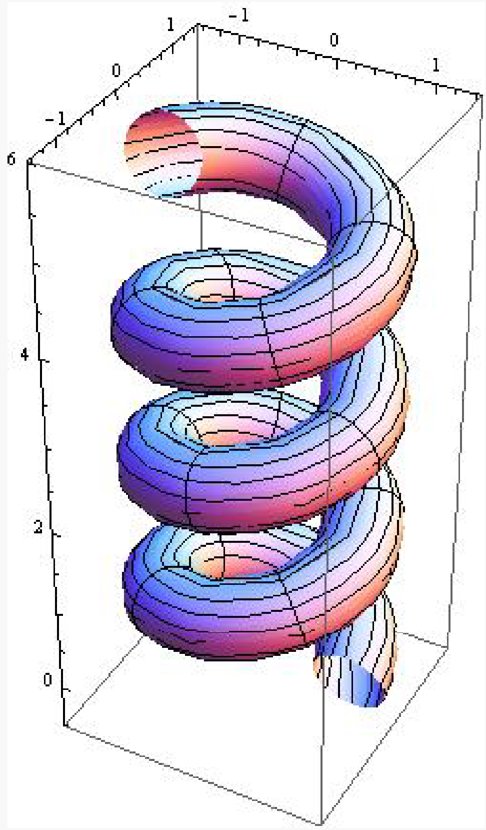
Nel nostro caso l'elica base da cui siamo partiti a ragionale veniva espressa dalla seguente equazione

```
elicaBase=
ParametricPlot3D[elica[1,0.25][t],{t,0,6 Pi}]
```

L'**ELICOIDE** invece è *una superficie enerata dal movimento rigido elicoidale di una curva.*
In particolare si ha un elicoide rigato quando la curva in questione è una retta; se tale retta si dispone perpendicolarmente all'asse si dirà Elicoide retto.

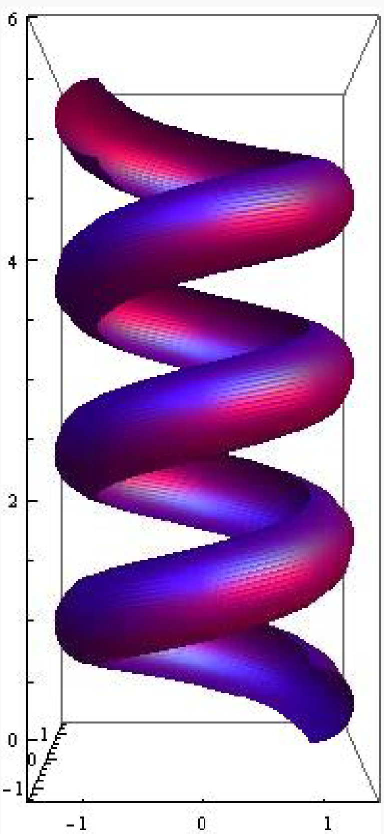
Il Tubo

Seguendo l'andamento dell'elica, la superficie denominata "tubo" ripropone il profilo della sezione ad arco della decorazione, cercando, con la variazione di alcuni parametri, di rispecchiare il più possibile le proporzioni della colonna studiata.



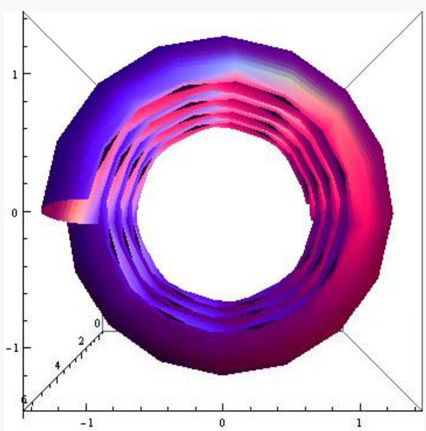
```
tubo[gamma_][r_][t_,teta_]:=
gamma[t]+r(Cos[teta]*normal[gamma]
[t]+Sin[teta]*binormal[gamma][t])
```

```
T1=
ParametricPlot3D[tubo[elica[1,
0.25]][0.4][t,teta],{t,0,7 Pi},{teta,0,2 Pi}]
```



Dopo numerose prove, si è ottenuto l'opportuno profilo variando il dominio d'esistenza.

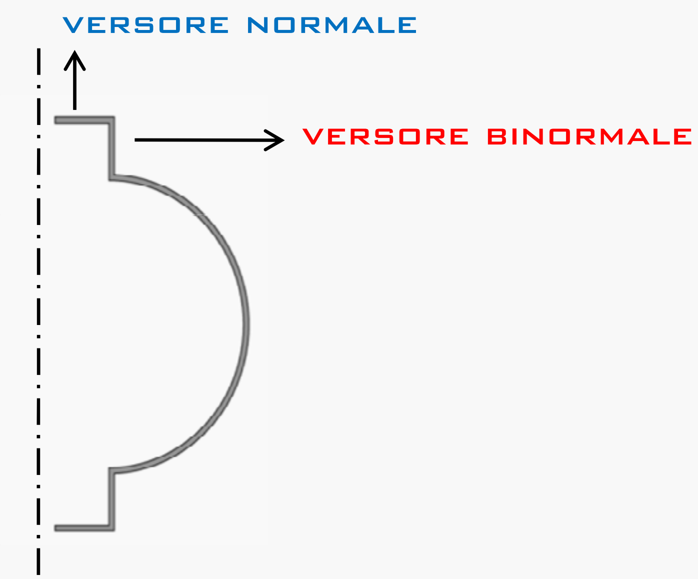
```
T2prospetto=
ParametricPlot3D[tubo[elica[1,0.25]][0.4][t
,teta],{t,0,7 Pi},{teta,Pi/2,3 Pi/2},ViewPoint-
>Front]
```

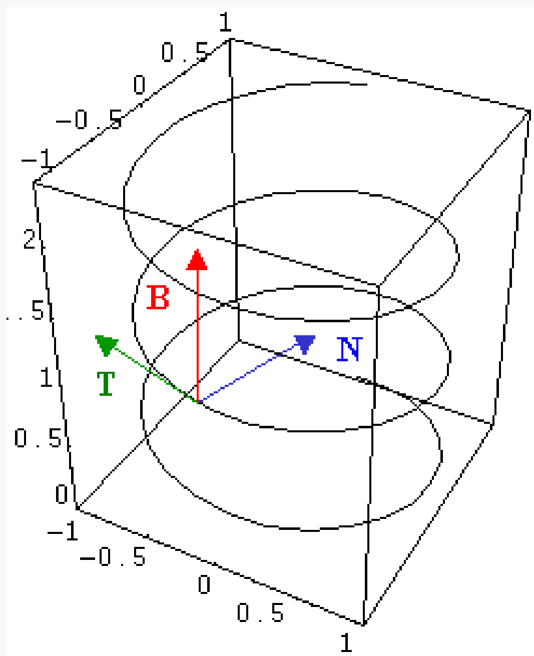


```
colonnaProspetto=
ParametricPlot3D
[tubo[elica[1,0.25]]
[0.4][t,teta],{t,0,7Pi},
{teta,Pi/2,3Pi/2},
PlotStyle->Directive
[Purple,Specularity
[White,20]],Mesh->
None,ViewPoint->Front]
```

Le funzioni utili

```
tangent[gamma_][t_]:=
Simplify[D[gamma[tt],tt]/(D[gamma[tt],tt].D[gamma[tt],tt])^(1/2)] /. Tt->t

normal[gamma_][t_]:=
Simplify[Cross[binormal[gamma][tt],tangent[gamma][tt]]] /. Tt->t

binormal[gamma_][t_]:=
Simplify[Cross[D[gamma[tt],tt],
D[gamma[tt],{tt,2}]]/(Cross[D[gamma[tt],tt],D[gamma[tt],{tt,2}]].
Cross[D[gamma[tt],tt],D[gamma[tt],{tt,2}]])^(1/2)] /. Tt->t

curvatura[gamma_][t_]:=
Simplify[(Cross[D[gamma[tt],tt],D[gamma[tt],{tt,2}]].
Cross[D[gamma[tt],tt],D[gamma[tt],{tt,2}]])^(1/2)/(D[gamma[tt],tt].
D[gamma[tt],tt])^(3/2)] /. Tt->t

torsione[gamma_][t_]:=
Simplify[(Cross[D[gamma[tt],tt],D[gamma[tt],{tt,2}]].
D[gamma[tt],{tt,3}])/(Cross[D[gamma[tt],tt],D[gamma[tt],{tt,2}]].
Cross[D[gamma[tt],tt],D[gamma[tt],{tt,2}]])] /. Tt->t
```





I Piani

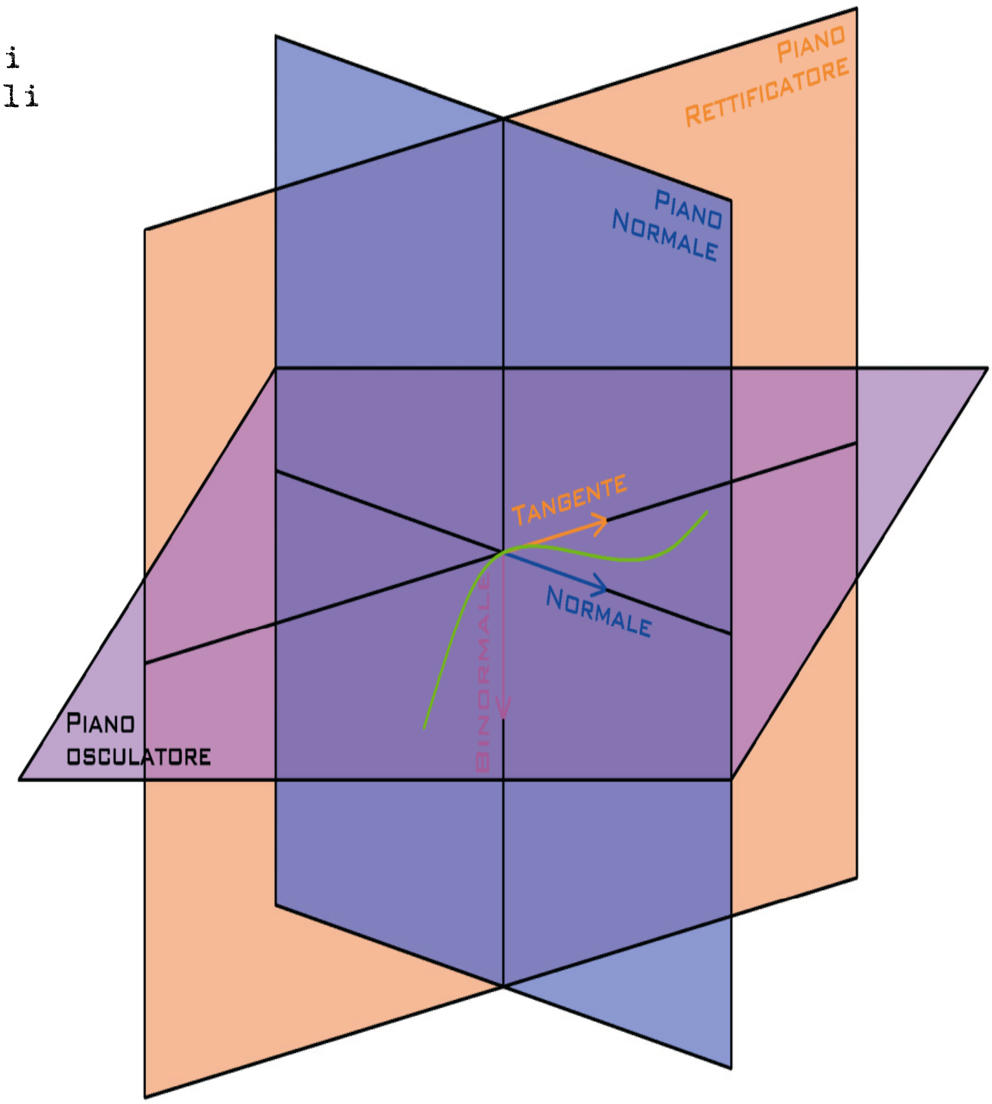
Come visto precedentemente, abbiamo avuto bisogno di caricare nel file di Mathematica, alcune funzioni che interverranno successivamente e che saranno quindi incluse esplicitamente o implicitamente nelle nostre specifiche funzioni.

Si è qui cercato di riportare brevemente degli accenni di geometria, utili a ricordarci concetti sui vettori in questione.

-RETTA TANGENTE = RETTA CONTENENTE IL VERSORE TANGENTE T

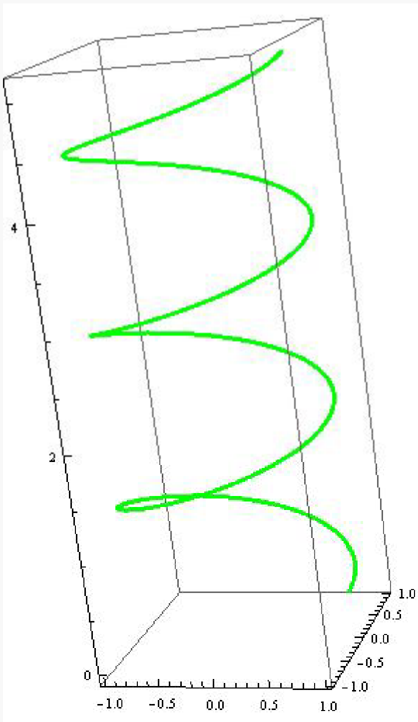
-RETTA NORMALE = RETTA CONTENENTE IL VERSORE NORMALE N

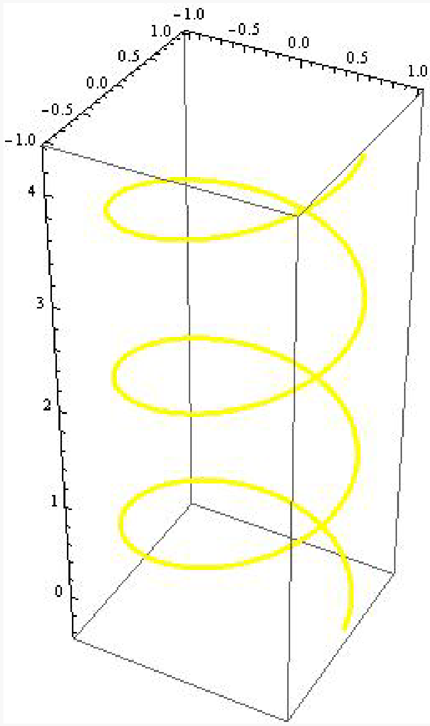
-RETTA BINORMALE = RETTA CONTENENTE IL VERSORE BINORMALE B

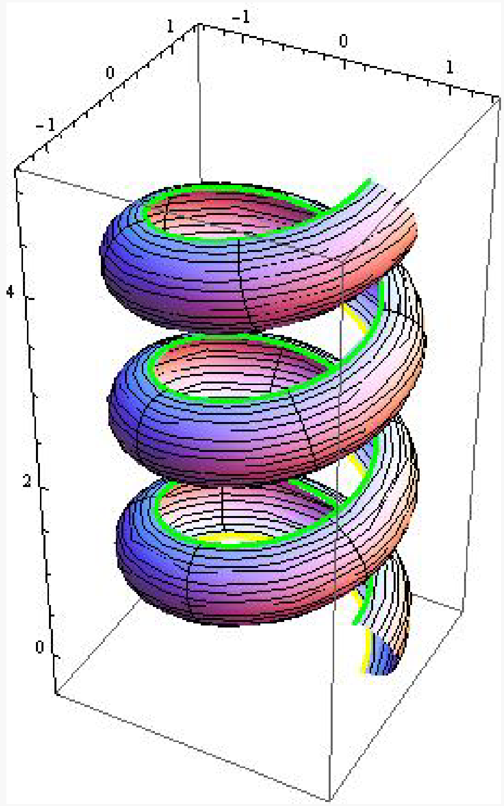


- PIANO RETTIFICATORE = PIANO TANGENTE/BINORMALE TB
- PIANO NORMALE = PIANO NORMALE/BINORMALE NB
- PIANO OSCULATORE = PIANO TANGENTE NORMALE TN

Curve "note"







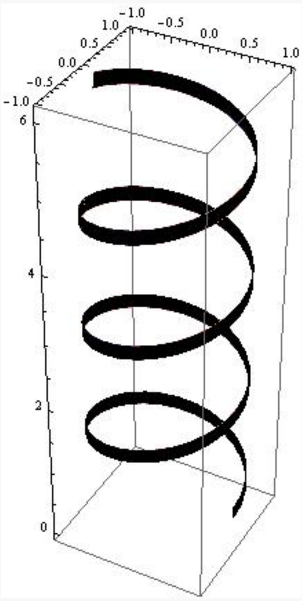
Evidenziamo le curve di importanza maggiore, dalle quali, come vedremo avranno origine le superfici normali e binormali.

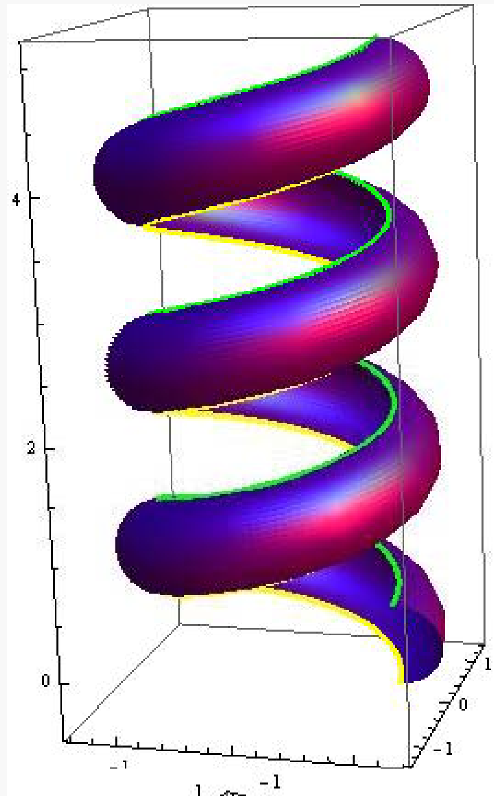
```
curvasup=
ParametricPlot3D[Style[tubo[elica[1,1/4]][0.4][t,Pi/2],
{RGBColor[0,1,0],AbsoluteThickness[3]}],{t,0,7 Pi}]
```

```
curvainf=
ParametricPlot3D[Style[tubo[elica[1,0.25]][0.4][t,3Pi/2],
{RGBColor[1,1,0],AbsoluteThickness[3]}],{t,0,7 Pi}]
```

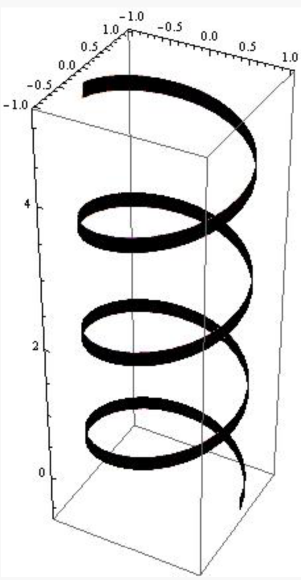
Normali e Binormali

Addizioniamo allora le componenti geometriche normali e binormali, per completare il profilo successivamente.

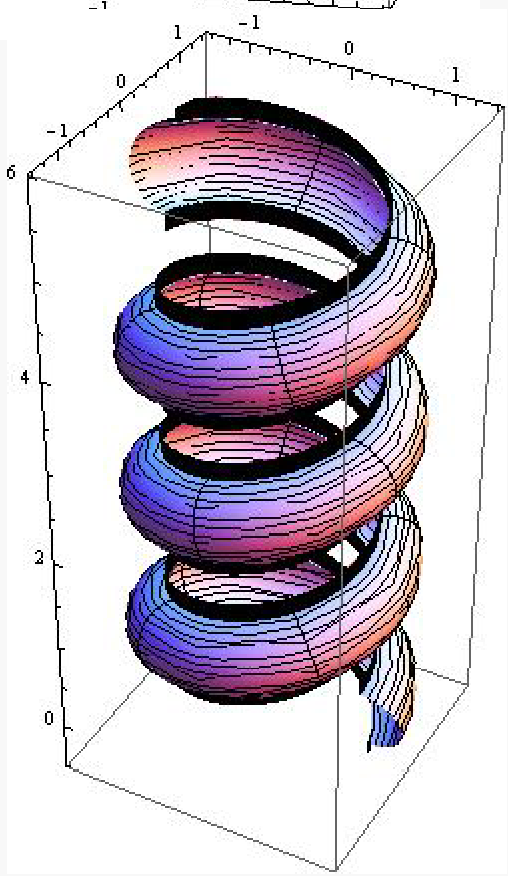




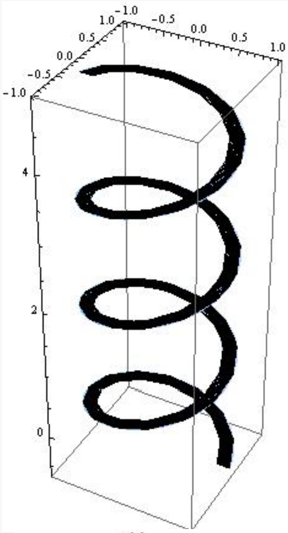
```
binormale1=
ParametricPlot3D
[tubo[elica[1,1/4]][0.4]
[t,3/2 Pi]teta*binormal
[curva2][t],
{t,0,7Pi},{teta,0,0.2},
PlotPoints->50]
```



```
binormale2=
ParametricPlot3D
[tubo[elica[1,1/4]][0.4]
[t,Pi/2]+teta*binormal
[curva2][t],
{t,0,7Pi},{teta,0,0.2},
PlotPoints->50]
```

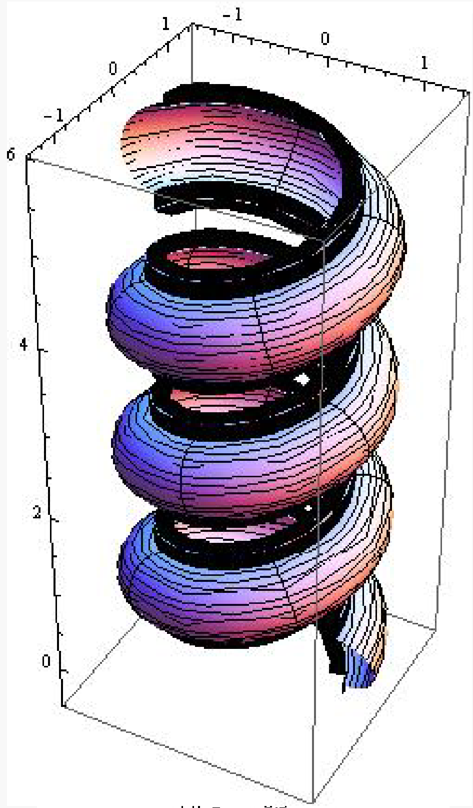


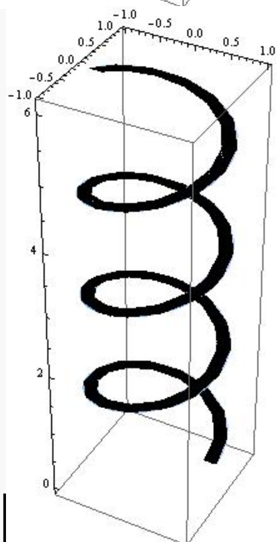
Binormale1 + Binormale2



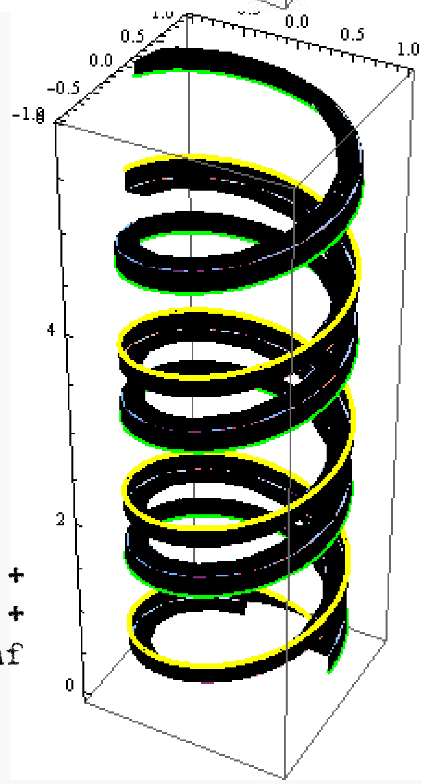
```
normale1=
ParametricPlot3D
[tubo[elica[1,1/4]][0.4]
[t,3/2 Pi]0.2*binormal
[curva2][t]+
teta*normal[curva2][t],
{t,0,7Pi},{teta,0,0.2},
PlotPoints->50]
```

Binormale1 + Binormale2 + normale1 + normale2





```
normale2=
ParametricPlot3D
[tubo[elica[1,1/4]][0.4]
[t,Pi/2]+0.2*binormal
[curva2][t]+teta*normal
[curva2][t],{t,0,7Pi},
{teta,0,0.2},
PlotPoints->50]
```

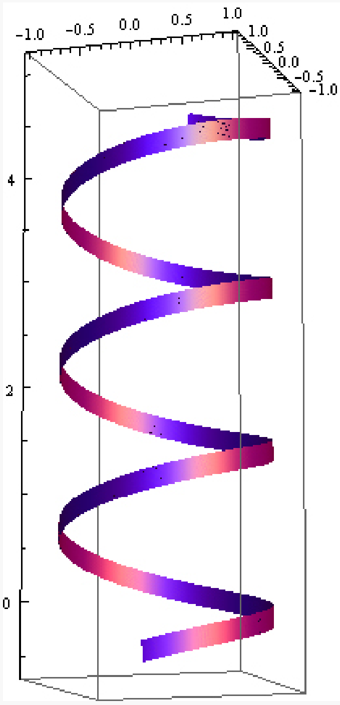


Si noti come nelle funzioni intervenga una "curva2" che sta ad indicare la necessità che tali geometrie hanno di essere maggiormente distaccate dal cilindro anima della funzione tubo.

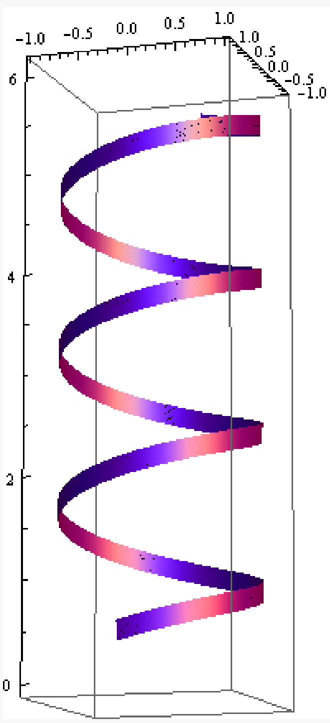
```
curva2[t_]:=
tubo[elica[3,1/4]][0.5][t,3/2 Pi]
```

Binormale1 + Binormale2 + normale1 + normale2 + curvasup + curvainf

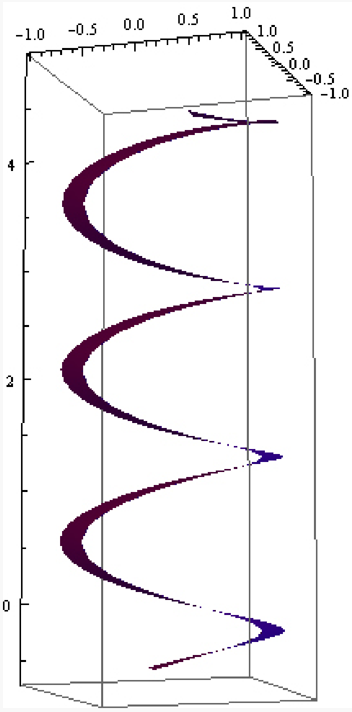
Riportiamo ora tutte le superfici con effetti cromatici realizzati appositamente per il render finale



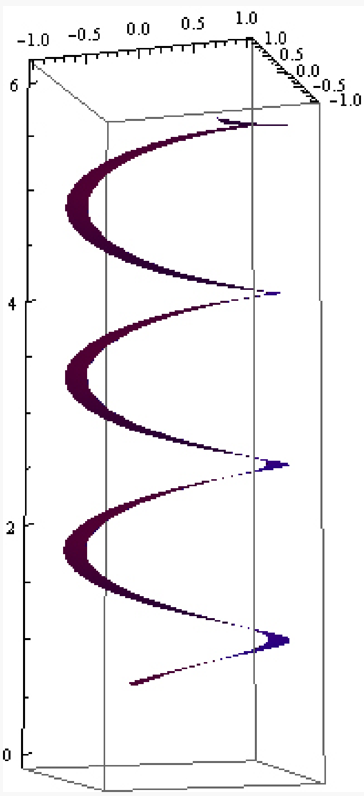
```
binormale1Render=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,3/2 Pi]-teta*Binormal[curva2][t],
{t,0,7Pi},{teta,0,0.2},PlotPoints->50,
PlotStyle->Directive[Purple,
Specularity[White,20]],
Mesh->None,ViewPoint->{3 Pi/2,Pi/2,-1}]
```



```
binormale2Render=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,Pi/2]+teta*Binormal[curva2][t],
{t,0,7Pi},{teta,0,0.2},PlotPoints->50,
PlotStyle->Directive[Purple,
Specularity[White,20]],
Mesh->None,ViewPoint->{3 Pi/2,Pi/2,-1}]
```




```
normale1Render=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,3/2 Pi]0.2*binormal[curva2][t]+
teta*normal[curva2][t],{t,0,7Pi},{teta,0,0.2},
PlotPoints->50,
PlotStyle>Directive[Purple,
Specularity[White,20]],Mesh>None,
ViewPoint->{3 Pi/2,Pi/2,-1}]
```

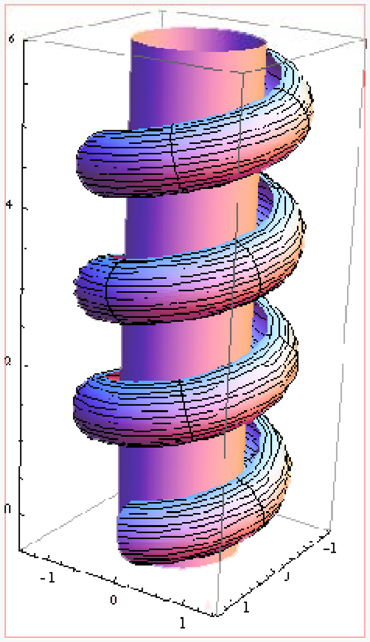


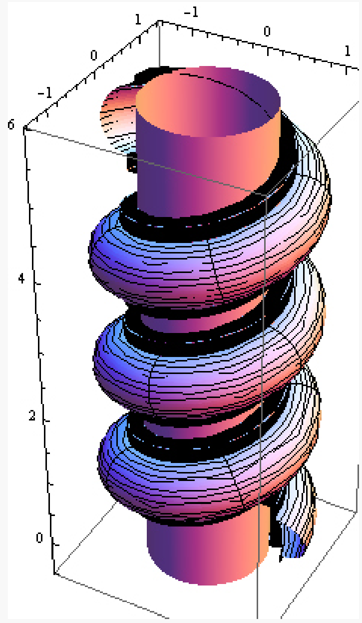

```
normale2Render=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,Pi/2 ]+0.2*binormal[curva2][t]+
teta*normal[curva2][t],{t,0,7Pi},{teta,0,0.2},
PlotPoints->50,
PlotStyle->Directive[Purple,
Specularity[White,20]],Mesh->None,
ViewPoint->{3 Pi/2,Pi/2,-1}]
```

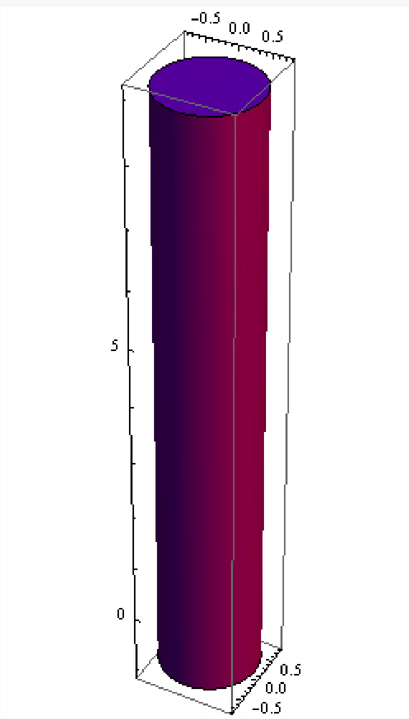
Il Cilindro

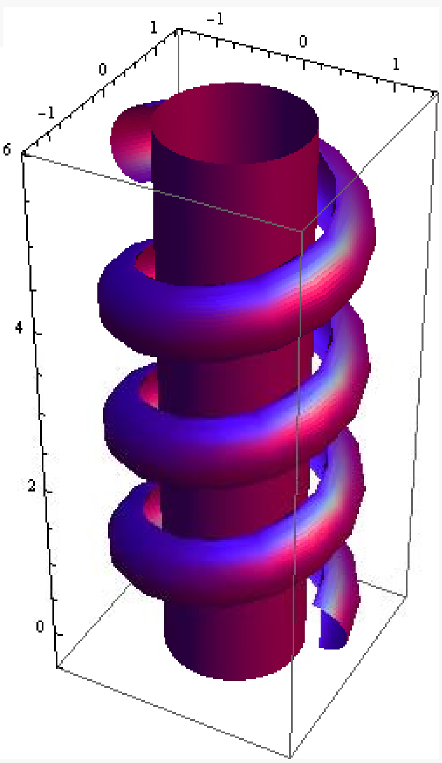
Infine non ci resta che caricare la funzione cilindro più adeguata.
Ricordiamo la questione, poco fa constatata, delle diverse curvature: più ampie per le superfici normali e binormali; più stretta per la sezione a toro.

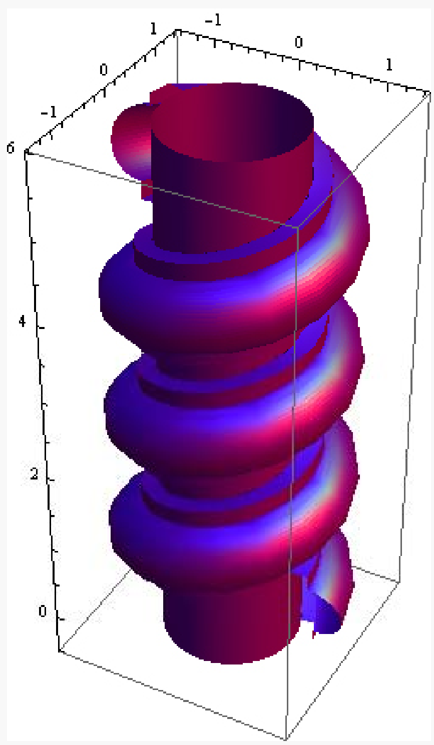
```
CilindroRender=
Graphics3D[{Purple,
Specularity[White,20],
Cylinder
[{{0,0,-1},{0,0,9}},0.8]},
Axes->True]
```



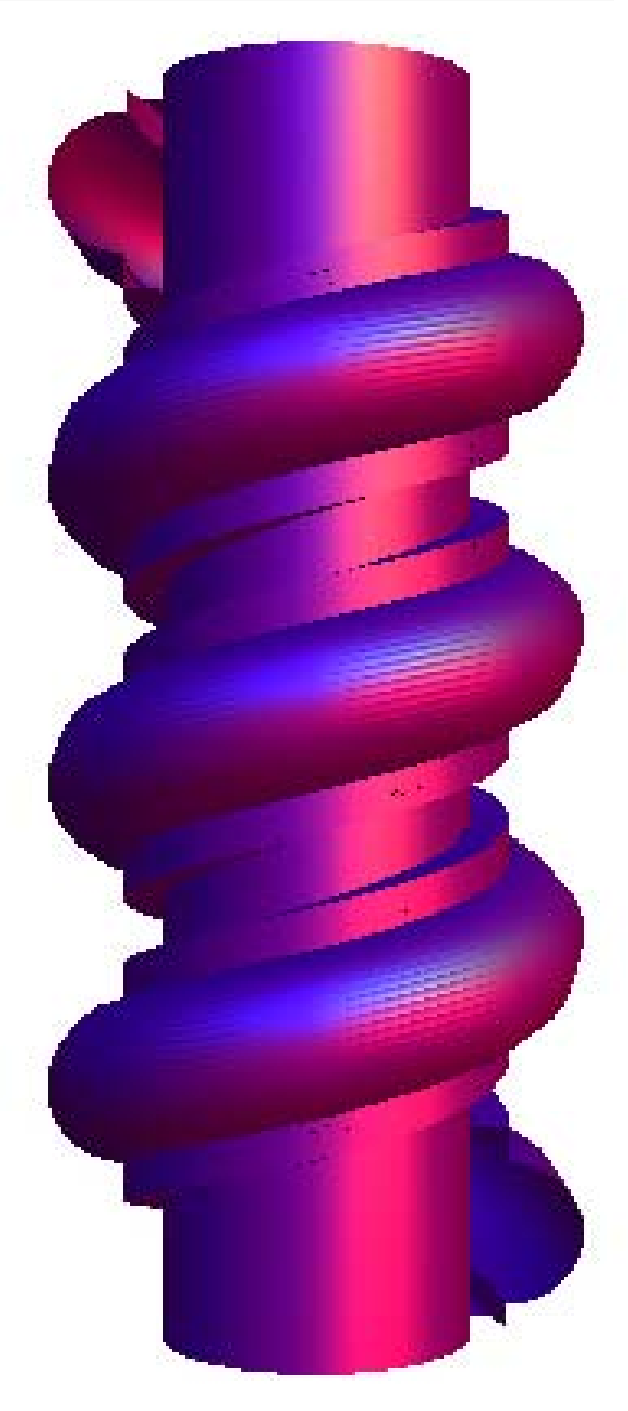


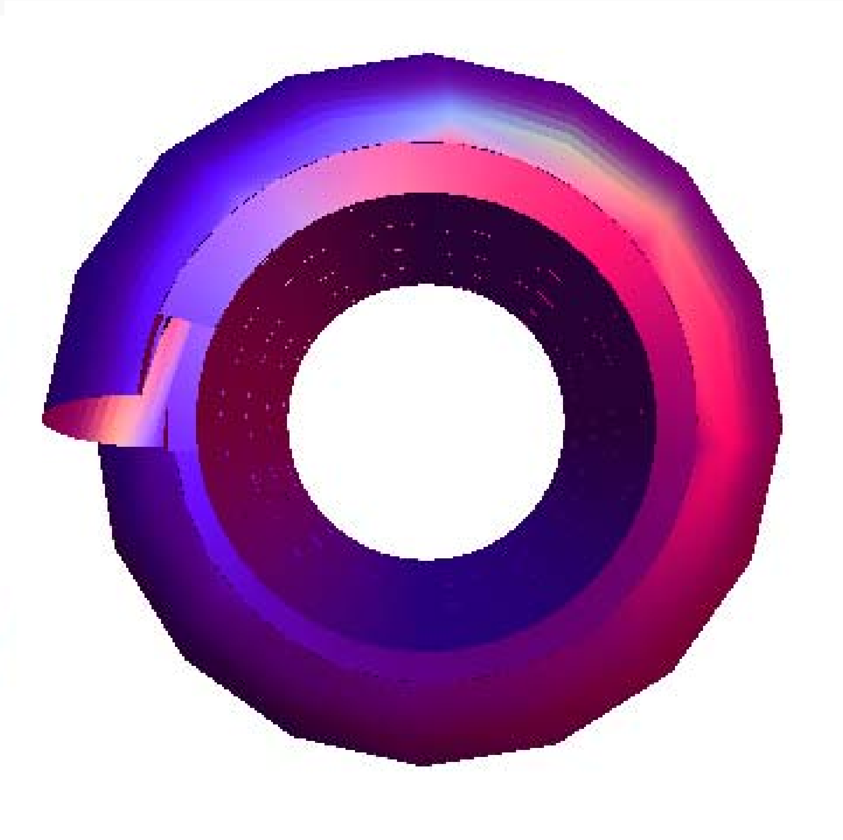






Render finale



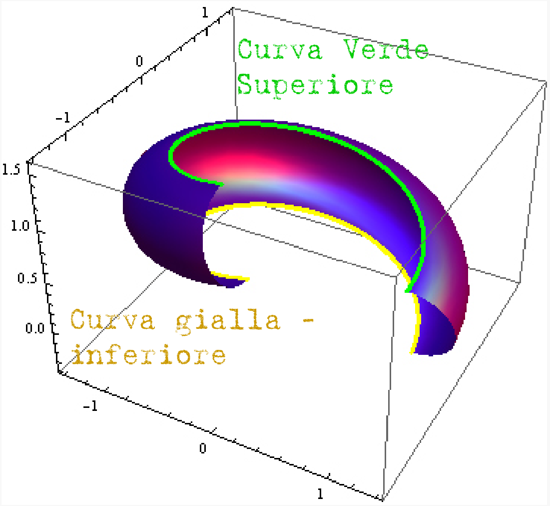


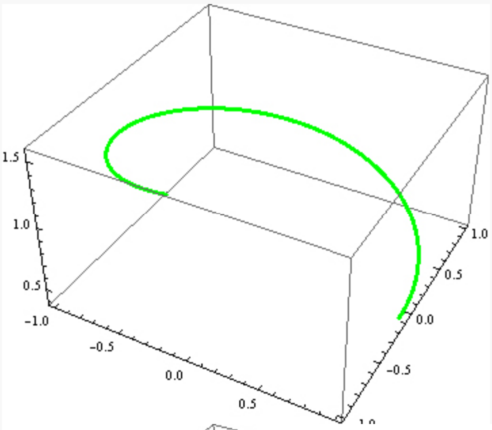
Elemento modulare - curve "note"

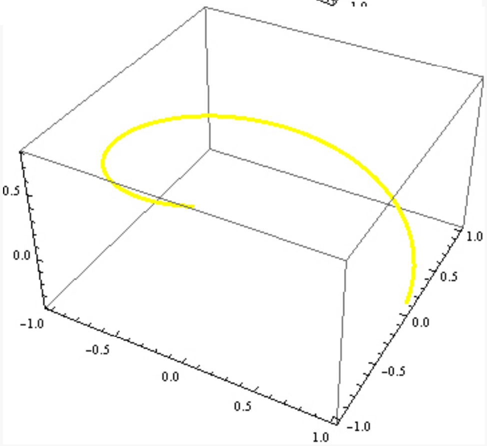
Lo studio di un solo tratto della superficie, grazie alla riduzione del PlotRange, ha permesso di aggiungere le porzioni di superficie normale e binormale che descrivono il profilo della decorazione.

Ciò si è ottenuto addizionando, all'espressione della superficie iniziale, un contributo nella direzione normale e binormale e aggiungendo dei coefficienti numerici per posizionarla spazialmente.

```
toro=
ParametricPlot3D
[tubo[elica[1,0.25]][0.4][t,teta],
{t,0,3 Pi/2},{teta,Pi/2,3 Pi/2},
ViewPoint->Front]
```



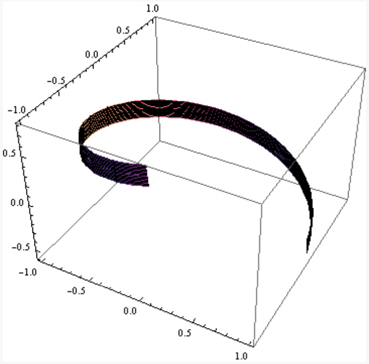




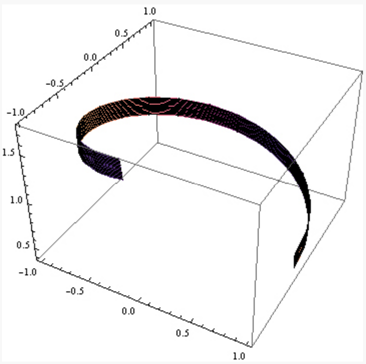
```
torocurvasup=
ParametricPlot3D
[Style[tubo[elica[1,1/4]][0.4][t,Pi/2],
{RGBColor[0,1,0],
AbsoluteThickness[3]}],{t,0,3 Pi/2}]
```

```
torocurvainf=
ParametricPlot3D
[Style[tubo[elica[1,0.25]][0.4]
[t,3Pi/2],{RGBColor[1,1,0],
AbsoluteThickness[3]}],{t,0,3 Pi/2}]
```

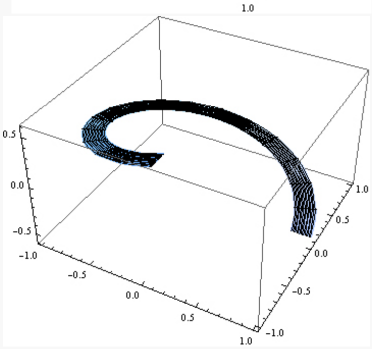
Elemento modulare - Normali e Binormali



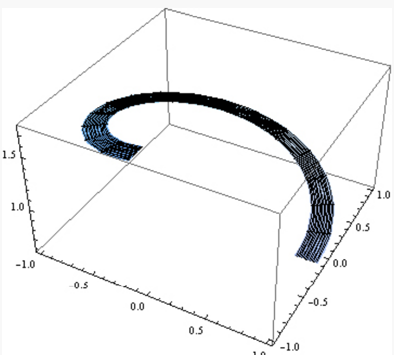
```
binormale1bis=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,3 Pi/2]-teta*binormal[curva2][t],
{t,0,3 Pi/2},{teta,0,0.2},PlotPoints->10]
```



```
binormale2bis=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,Pi/2 ]+teta*binormal[curva2][t],
{t,0,3 Pi/2},{teta,0,0.2},PlotPoints->10]
```



```
normale1bis=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,3/2 Pi]0.2*binormal[curva2][t]
+teta*normal[curva2][t],{t,0,3 Pi/2},
{teta,0,0.2},PlotPoints->10]
```

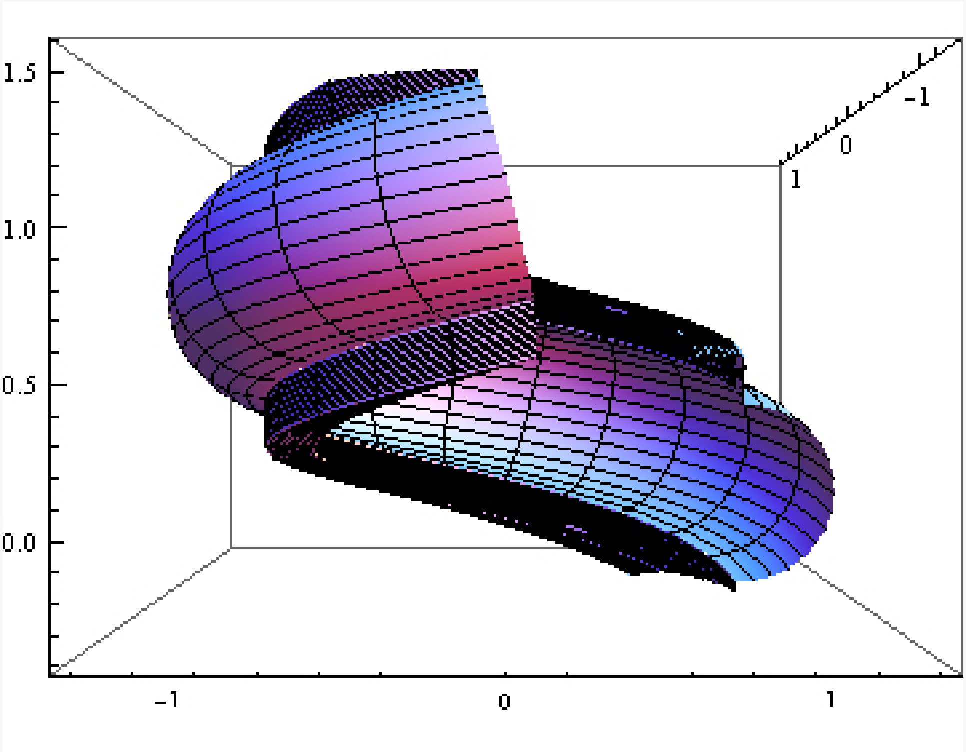


```
normale2bis=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,Pi/2 ]+0.2*binormal[curva2][t]
+teta*normal[curva2][t],
{t,0,3 Pi/2},{teta,0,0.2},PlotPoints->10]
```

Elemento modulare - Normali e Binormali

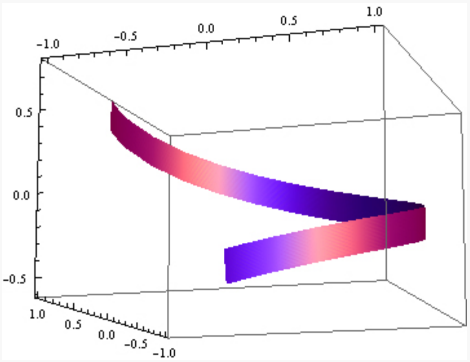
Ci troviamo dunque a ripetere le stesse operazioni svolte precedentemente, nel caso generico, applicandole ad un elemento solo modulare. Ciò è stato possibile riducendo i parametri interessati.

```
Show
[toro,binormale1bis,binormale2bis,normale1bis,normale2bis,
ViewPoint->Front]
```

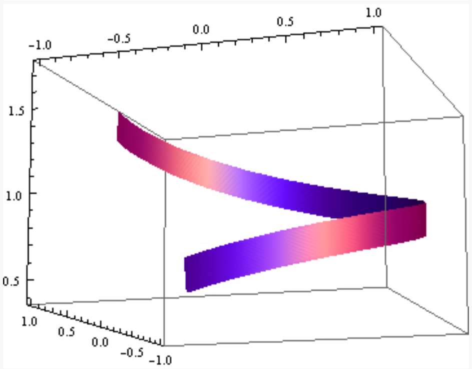


Profilo di Decorazione

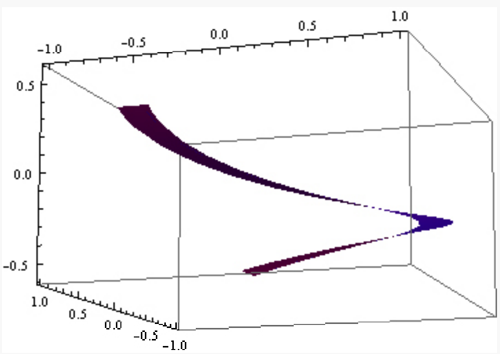
Riportiamo ora tutte le superfici con effetti cromatici realizzati appositamente per il render finale



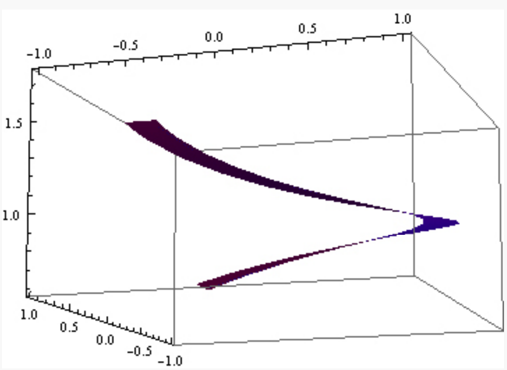

```
binormale1bisRender=
ParametricPlot3D[tubo[elica[1,1/4]]
[0.4][t,3/2 Pi]teta*Binormal[curva2][t],
{t,0,3Pi/2},{teta,0,0.2},PlotPoints->10,
PlotStyle->Directive[Purple,
Specularity[White,20]],Mesh->None,
ViewPoint->{3 Pi/2,Pi/2,-1}]
```




```
binormale2bisRender=
ParametricPlot3D[tubo[elica[1,1/4]]
[0.4][t,Pi/2 ]+teta*Binormal[curva2][t],
{t,0,3Pi/2},{teta,0,0.2},PlotPoints->10,
PlotStyle->Directive[Purple,
Specularity[White,20]],Mesh->None,
ViewPoint->{3 Pi/2,Pi/2,-1}]
```

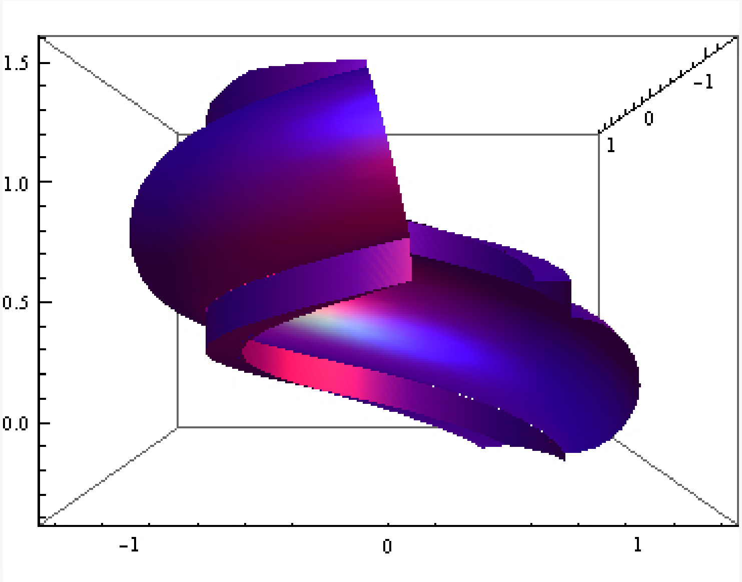


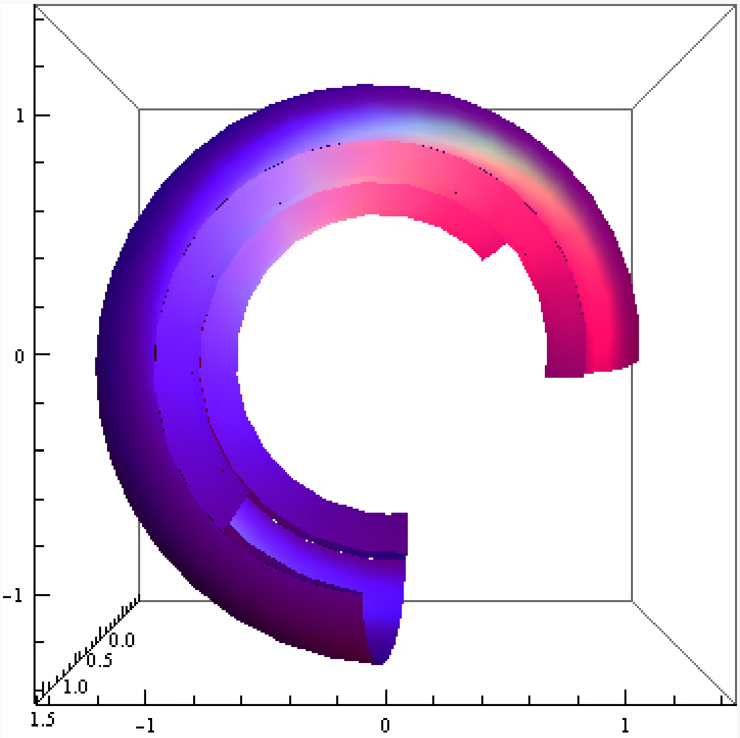

```
normale1bisRender=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,3/2 Pi]0.2*Binormal[curva2][t]
+teta*Normal[curva2][t],{t,0,3Pi/2},
{teta,0,0.2},PlotPoints->10,
PlotStyle->Directive[Purple,
Specularity[White,20]],Mesh->None,
ViewPoint->{3 Pi/2,Pi/2,-1}]
```

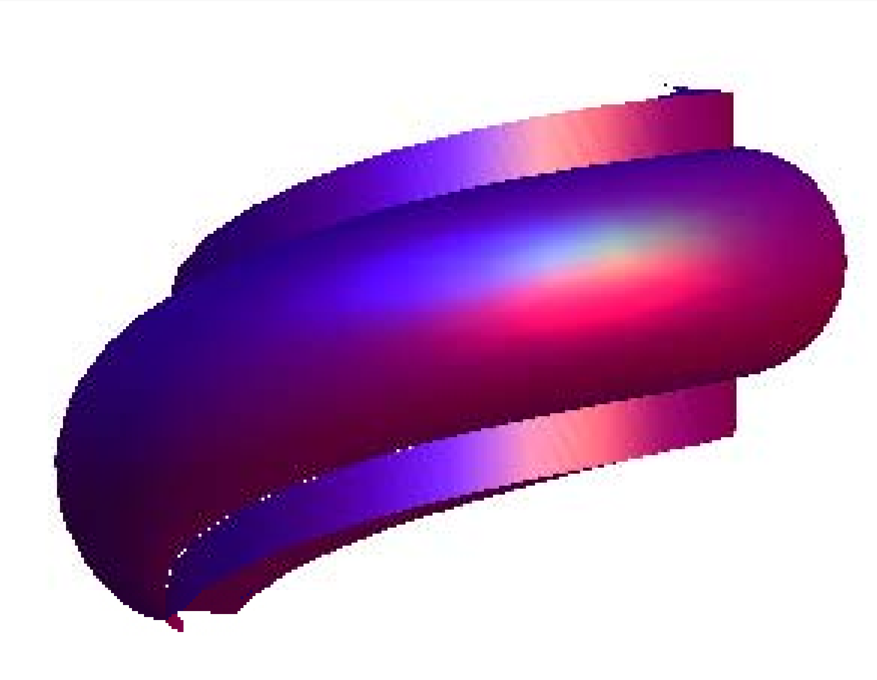



```
normale2bisRender=
ParametricPlot3D[tubo[elica[1,1/4]][0.4]
[t,Pi/2]+0.2*Binormal[curva2][t]+
teta*Normal[curva2][t],{t,0,3Pi/2},
{teta,0,0.2},PlotPoints->10,
PlotStyle->Directive[Purple,
Specularity[White,20]],Mesh->None,
ViewPoint->{3 Pi/2,Pi/2,-1}]
```

Render finale







Bibliografia

- Marco Bussagli, *Capire l'architettura*, Firenze 2003
- Ciol Elio; Ciol Stefano, *La facciata del Duomo di Orvieto. Teologia in figura*, Milano 2002
- G.B. Thomas; R.L. Finney, *Analisi Matematica*, ed. Zanichelli
- Salsa; Squellati, *Calcolo infinitesimale e algebra lineare*
- P. Adorno, *L'arte italiana*, Vol. 1
- Sito internet ufficiale del Duomo di Orvieto, *http://www.opsm.it/duomo/001.html*
- Lezioni in aula e Materiale didattico del Prof. Falcolini (a.a. 2007/2008)