9.3 Τρισδιάστατες Γραφικές Παραστάσεις

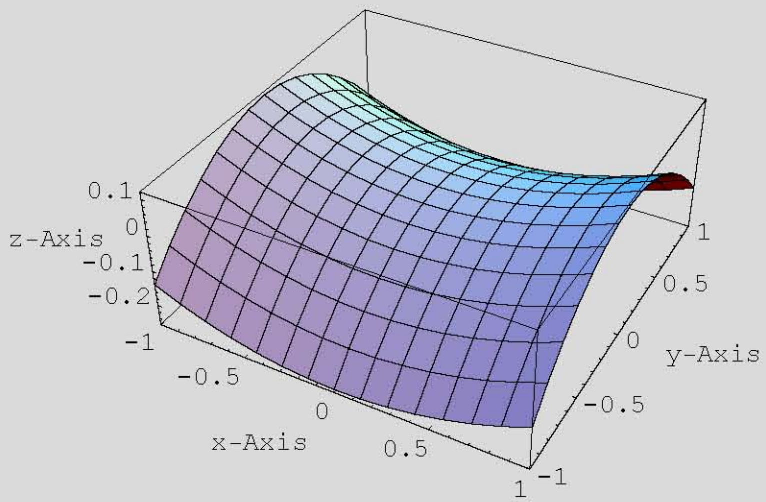
Υπάρχει αρκετή ομοιότητα στις εντολές και στις επιλογές για την γραφική παράσταση επιφανειών και καμπυλών στο χώρο, με τις αντίστοιχες που γνωρίσαμε στα διδιάστατα γραφικά. Έτσι σε πολλές περιπτώσεις δεν θα αναφέρουμε πολλές λεπτομέρειες. Γι'αυτό θα συνιστούσαμε να διαβάσετε ξανά το κεφάλαιο για τις διδιάστατες γραφικές παραστάσεις, και να ανακαλύψετε τις ομοιότητες και τις διαφορές που υπάρχουν μεταξύ των εντολών.

9.3.1 Γραφικές παραστάσεις επιφανειών.

Χρησιμοποιούμε διαφορετικές εντολές ανάλογα με τον τρόπο που περιγράφεται η επιφάνεια που μελετάμε. Έτσι έχουμε τις παρακάτω περιπτώσεις:

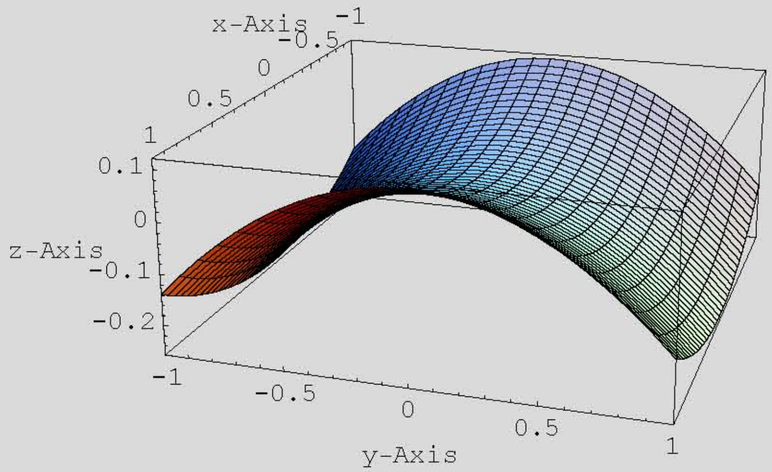
1. Η επιφάνεια είναι το **γράφημα μιας συνάρτησης** δυο μεταβλητών(π.χ της x και y). Τότε χρησιμοποιούμε την Plot3D π.χ εαν $f[x, y] = \frac{x^2}{9} - \frac{y^2}{4}$και πεδίο ορισμού το [-1,1]X[-1,1] τότε παίρνουμε την εντολή

```
Plot3D[x^2 / 9 - y^2 / 4, {x, -1, 1}, {y, -1, 1},
  AxesLabel → {"x-Axis", "y-Axis", "z-Axis"}];
```



Προσέξτε ότι οι άξονες δεν σχεδιάζονται έτσι ώστε να έχουν κοινό σημείο την αρχή (0,0,0) αλλά βρίσκονται πάνω σε κάποιες πλευρές του κουτιού (Box) που περιβάλλει την επιφάνεια. Επίσης προσέξτε ότι το σύστημα αξόνων είναι δεξιόστροφο. Είναι δύσκολο να εντοπίσετε που βρίσκεται το Oxy επίπεδο, που το Oxz και που το Oyz. Θα πρέπει να εντοπίσετε κατ' αρχήν που βρίσκεται στο χώρο το σημείο (0,0,0). Το Oxy είναι εκείνο το επίπεδο που περνάει απο το παραπάνω σημείο και είναι κάθετο στον κάθετο άξονα " z-Axis". Επίσης ο άξονας Ox βρίσκεται παράλληλος προς εμάς. Εαν έχετε συνηθίσει να βλέπετε τον άξονα Ox να έρχεται κάθετα προς εσάς τότε πρεπει να αλλάξετε την γωνία θέασης του σχήματος(ViewPoint) και να θέσετε για παράδειγμα ViewPoint->{3.055, 0.947, 1.104}. Για το ViewPoint θα μιλήσουμε παρακάτω. Επίσης είναι σημαντικό να παρατηρείσετε ότι η μονάδα μέτρησης στον Oz είναι μικρότερη απο τις άλλες στους άλλους άξονες. Αυτό οφείλεται στην προεπιλεγμένη τιμή BoxRatios→{1,1,0.4}της Plot3D. Για αυτό το θέμα θα μιλήσουμε ξανά παρακάτω. Η επιφάνεια αυτή δεν είναι άλλη απο το υπερβολικό παραβολοειδές δηλ. το "σαμάρι". Παρατηρούμε ότι η επιφάνεια έχει καλυφθεί απο καμπύλες παράλληλες με τους άξονες Ox και Oy. Αυτές τέμνονται κάθετα και έτσι σχηματίζεται ένα πλέγμα(Mesh). Ουσιαστικά επιλέγονται 15 σημεία σε κάθε κατεύθυνση οπότε έχουμε 15X15 σημεία πάνω στο επίπεδο Oxy! Αυτά είναι τα δειγματοληπτικά σημεία(PlotPoints) δηλ. σε κάθε ένα απο αυτά υπολογίζεται η τιμή της συνάρτησης f[x,y] και στην συνέχεια με βάση αυτές τις τιμές, σχεδιάζεται το πλέγμα και τελικά η ίδια η επιφάνεια . Είναι λοιπόν ευνόητο αν θέλουμε καλύτερη ακρίβεια στην γραφική παράσταση να πάρουμε περισσότερα PlotPoints π.χ με PlotPoint->{40,25}διαλέγουμε 40 σημεία (απο το πεδίου ορισμού της f) πάνω στο Ox και 25 πάνω στον Oy(πάλι απο το πεδίου ορισμού της f). Π.χ

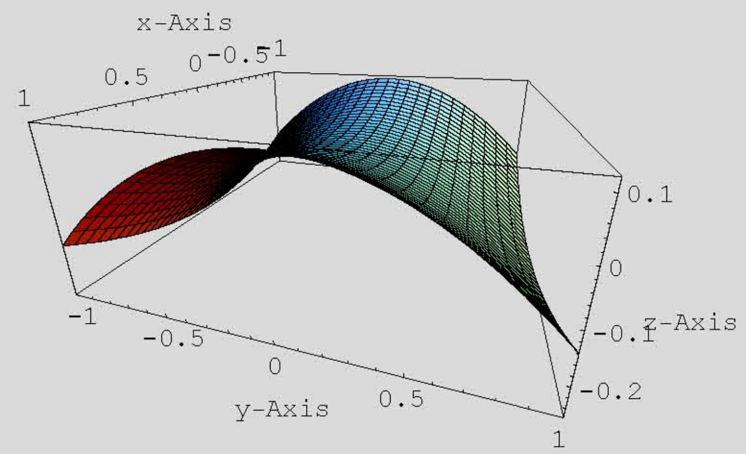
```
Plot3D[x^2/9 - y^2/4, {x, -1, 1}, {y, -1, 1},
  PlotPoints → {40, 25}, ViewPoint → {3.055, 0.947, 1.104},
  AxesLabel → {"x-Axis", "y-Axis", "z-Axis"}];
```



Εδώ αλλάξαμε και την γωνία θέασης γιαυτό ίσως σας φαίνεται διαφορετικό απο το προηγούμενο. Με αυτήν την γωνία θέασης οι άξονες έχουν τοποθετηθεί με την διάταξη που συνήθως ξέρουμε δηλ. ό άξονας O x να έρχεται πρός εμάς. Με AxesLabel→{"x-Axis", "y-Axis", "z-Axis"} θέσαμε ετικέτες στους άξονες για να ξέρουμε ποιοί είναι...

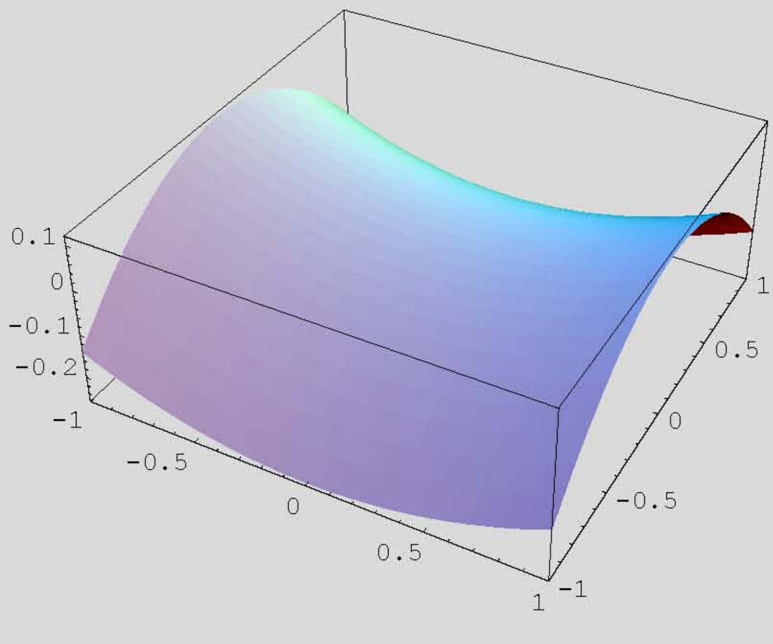
Παρατήρηση. Αν αλλάξετε την γωνία θέασης παίρνοντας για παράδειγμα μικρότερες τιμές στο ViewPoint τότε το γράφημα θα σας βγεί παραμορφωμένο και όπως θα το έβλεπε ένας παρατηρητής που βρισκόταν πολύ κοντά. Επίσης είναι απαραίτητο το ViewPoint να είναι έξω απο το(τις συντεταγμένες του)κουτί που περιβάλλει το γράφημα ..Π.χ

```
Plot3D[x^2 / 9 - y^2 / 4, {x, -1, 1}, {y, -1, 1},
  PlotPoints → {40, 25}, ViewPoint → {3.055 / 3, 0.947 / 3, 1.104 / 3},
  AxesLabel → {"x-Axis", "y-Axis", "z-Axis"}];
```



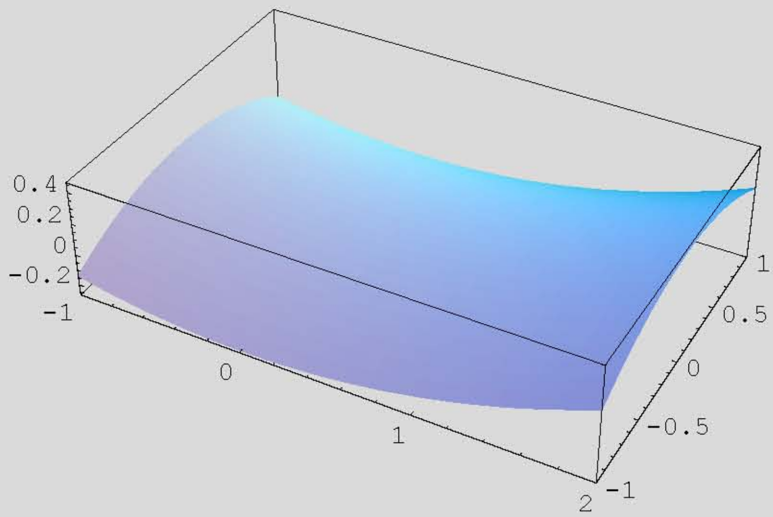
Με Mesh->False σχεδιάζεται η επιφάνεια μας χωρίς να είναι εμφανές το πλέγμα πάνω σ'αυτήν.

```
Plot3D[x^2 / 9 - y^2 / 4, {x, -1, 1},
  {y, -1, 1}, PlotPoints → {40, 25}, Mesh → False];
```



Το διάφανο κουτί(Box) που περιβάλλει το πλέγμα, έχει λόγο πλευρών {1,1,0.4} δηλ. το μήκος του μήκους και του πλάτους είναι το ίδιο ενώ το ύψος του κουτιού είναι μικρότερο κατά 0.4 δηλ. κατα 40%. Αυτή είναι η προεπιλεγμένη τιμή του BoxRatios και μπορείτε να την αλλάξετε! Π.χ να θέσετε BoxRatios->{1,1,1} για να έχουν όλες οι πλευρές ίσο μήκος ή να θέσετε BoxRatios->Automatic π.χ

```
Plot3D[x^2/9 - y^2/4, {x, -1, 2}, {y, -1, 1},
  PlotPoints → {40, 25}, Mesh → False, BoxRatios → Automatic];
```



Με BoxRatios->Automatic έχουμε ίσες μονάδες μέτρησης σε κάθε άξονα και άρα το "φυσιολόγικό" γράφημα της επιφάνειας. Αυτό βέβαια δεν είναι πάντα επιθυμητό γιατί χάνουμε κάποια χαρακτηριστικά της επιφάνειας. Π.χ δεν φαίνεται καθαρά το "σαμαρωτό" σχήμα της επιφάνειας. Με BoxRatios->{1,1,2.5} θα έχετε καλύτερο αποτέλεσμα, δοκιμάστε!

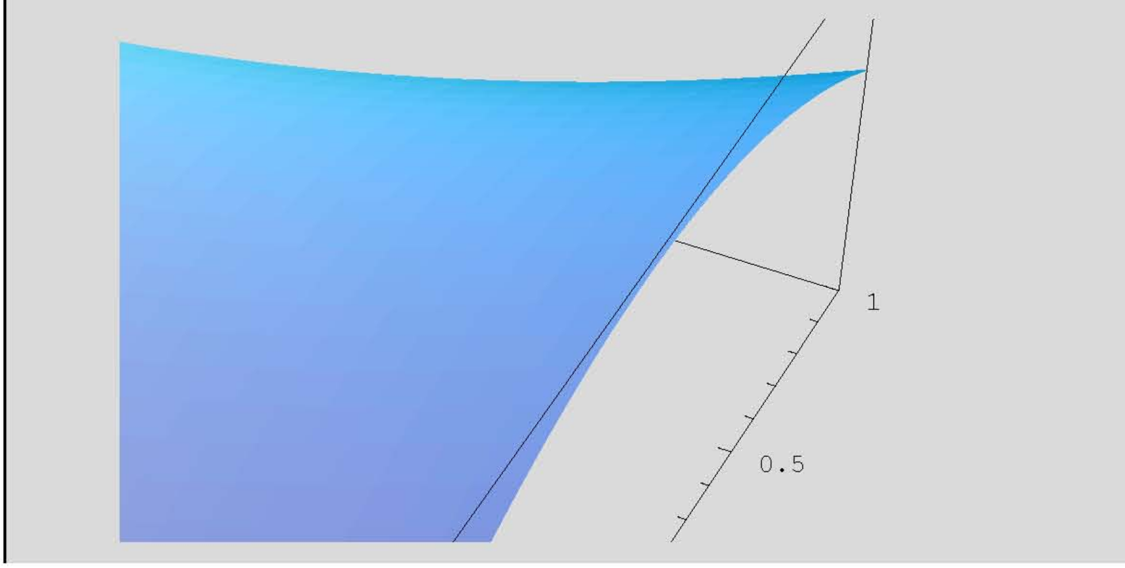
Μια πολύ χρήσιμη εντολή είναι η **Options[..]** διότι μπορούμε να διαπιστώσουμε τις επιλογές μιας εντολής καθώς και τις προεπιλεγμένες τιμές των επιλογών π.χ

```
Options[ParametricPlot3D]

{AmbientLight → GrayLevel[0.], AspectRatio → Automatic,
 Axes → True, AxesEdge → Automatic, AxesLabel → None,
 AxesStyle → Automatic, Background → Automatic, Boxed → True,
 BoxRatios → Automatic, BoxStyle → Automatic, ColorOutput → Automatic,
 Compiled → True, DefaultColor → Automatic, Epilog → {},
 FaceGrids → None, ImageSize → Automatic, Lighting → True,
 LightSources → {{{1., 0., 1.}, RGBColor[1, 0, 0]}, {{1., 1., 1.},
    RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}},
 Plot3Matrix → Automatic, PlotLabel → None, PlotPoints → Automatic,
 PlotRange → Automatic, PlotRegion → Automatic,
 PolygonIntersections → True, Prolog → {}, RenderAll → True,
 Shading → True, SphericalRegion → False, Ticks → Automatic,
 ViewCenter → Automatic, ViewPoint → {1.3, -2.4, 2.},
 ViewVertical → {0., 0., 1.}, DefaultFont :→ $DefaultFont,
 DisplayFunction :→ $DisplayFunction,
 FormatType :→ $FormatType, TextStyle :→ $TextStyle}
```

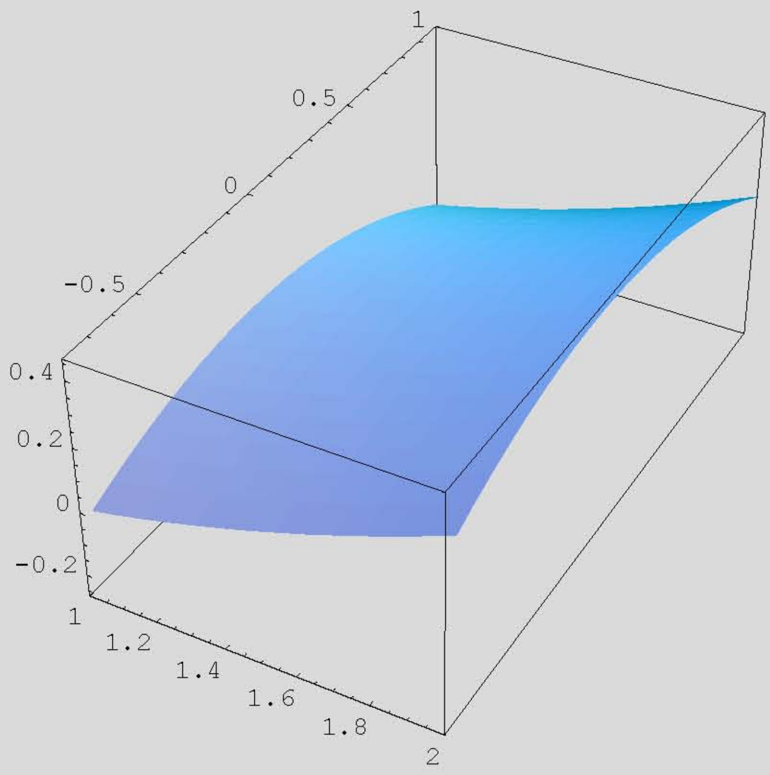
Άσκηση: Μελετήστε τις διαφορές των options στις Plot και Plot3D. Παρατηρείστε για παράδειγμα ότι στην δεύτερη συνάρτηση δεν υπάρχει η επιλογή PlotStyle, Frame, ή AxesOrigin . Ομως υπάρχει η AxesEdge, η ViewCenter και η ViewVertical. Μάθετε τι δουλειά κάνουν με την βοήθεια του Help. Πειραματιστείτε και με τις άλλες επιλογές. Όπως γνωρίζουμε ήδη με κατάλληλες αρνητικές τιμές του PlotRegion μπορούμε και να "ζουμάρουμε ". Δοκιμάστε το

```
Plot3D[x^2 / 9 - y^2 / 4, {x, -1, 2},
  {y, -1, 1}, PlotPoints → {40, 25}, Mesh → False,
  BoxRatios → Automatic, PlotRegion → {{-2, 1}, {-3, 4}}];
```



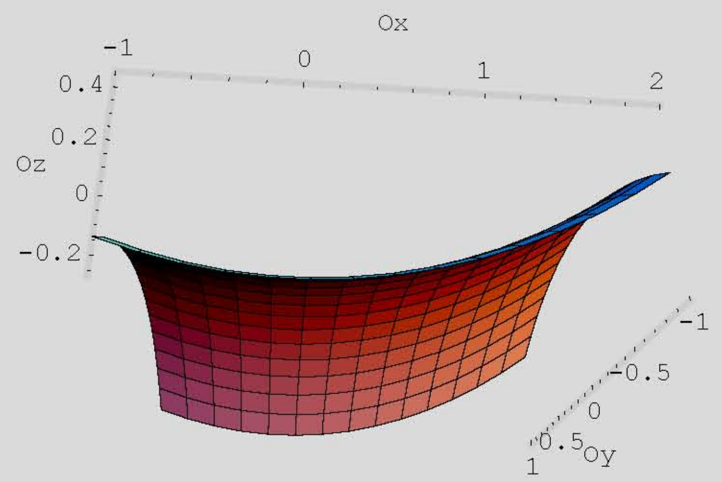
Μπορείτε να ζουμάρετε και αν πάρετε μικρότερο PlotRange απ΄ότι το πεδίο ορισμού π.χ

```
Plot3D[x^2 / 9 - y^2 / 4, {x, -1, 2},
  {y, -1, 1}, PlotRange → {{1, 2}, {-0.8, 1}, All},
  BoxRatios → Automatic, Mesh → False, PlotPoints → {40, 25}];
```



Παρατηρούμε ότι έχουμε πάρα πολλές δυνατότητες. Βέβαια αρκετές απ' αυτές τις έχουμε δει ξανά στην διδιάστατη περίπτωση και έτσι δεν θα αναφέρουμε περισσότερα. Τέτοιες είναι οι MeshStyle, Background, Epilog, ImageSize,PlotLabel,PlotRange,PlotRegion, DisplayFunction, AxesLabel, και Ticks. Θα αναφέρουμε την **View-Point**. Με την ViewPoint μπορούμε να δούμε με διαφορετική οπτική γωνία την ίδια επιφάνεια. Με τα πλήκτρα Ctrl Shift και V συγχρόνως πατημένα, παίρνουμε ένα κουτί με τους τρεις άξονες στο οποίο μπορούμε να επιλέξουμε το κατάλληλο σημείο(ViewPoint) του χώρου(απο το οποίο θα παρατηρούμε την επιφάνεια) και πατώντας Paste μπορούμε να το επικολήσουμε σε όποιο σημείο της Plot3D θέλουμε. Παρατηρείστε ότι η προεπιλεγμένη τιμή είναι ViewPoint->{1.3,-2.4,2.}. Με **Boxed->False** εξαφανίζουμε το διάφανο κουτί που περιβάλλει την επιφάνεια ενώ με **AxesStyle->...** μπορούμε να καθορίσουμε κάποιο "στύλ" με το οποίο θα σχδιαστούν οι τρείς κάθετοι άξονες. Παρακάτω σχεδιάζουμε το σαμάρι, εξαφανίζουμε το κουτί, θέτουμε Axes-Style→{GrayLevel[0.8], Thickness[0.01]}], θέτουμε επιγραφές σε κάθε άξονα και παρατηρούμε την επιφάνεια απο κάτω(ViewPoint->{-0.184, -1.819, -0.795}):

```
Plot3D[x^2 / 9 - y^2 / 4, {x, -1, 2}, {y, -1, 1},
  Boxed → False, AxesStyle → {GrayLevel[0.8], Thickness[0.01]},
  AxesLabel → {"Ox", "Oy", "Oz"},
  ViewPoint -> {-0.184, -1.819, -0.795}];
```



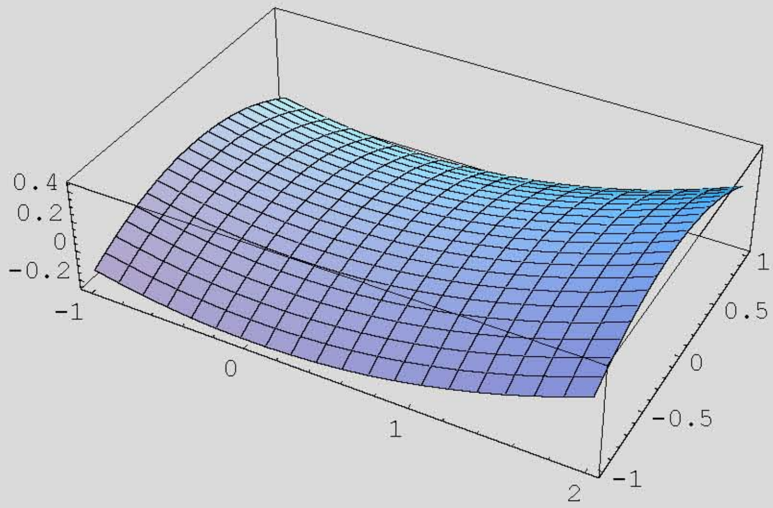
Με Ticks->None μπορούμε να εξαφανίσουμε τα Ticks που βρίσκονται πάνω στους άξονες ενώ με Axes-> False μπορούμε και να εξαφανίσουμε και τους ίδιους τους άξονες!

Άσκηση. Εξετάστε τις δυνατότητες του παρακάτω πακέτου Graphics`Animation`. Για παράδειγμα να τρέξετε και το πρόγραμμα που ακολουθεί.

```
<< Graphics`Animation`
SpinShow[Plot3D[x^2 / 9 - y^2 / 4, {x, -1, 1},
  {y, -1, 1}, PlotPoints → {40, 25}, Mesh → False]]
```

2. Τη γραφική παράσταση μιας επιφάνειας, που ορίζεται με την βοήθεια δύο παραμέτρων (π.χ u και v) την παίρνουμε με την **ParametricPlot3D**. Τέτοιες επιφάνειες λέγονται **παραμετρικές** επιφάνειες. Σε αυτήν την κατηγορία ανήκουν και επιφάνειες που ορίζονται με κυλινδρικές ή σφαιρικές συντεταγμένες. Αυτές οι επιφάνειες μπορεί να τέμνουν τον εαυτό τους ή να κουλουριάζουν γύρω απο κάποιο άξονα κ.ο.κ. Τέτοιες δυνατότητες δεν έχει η Plot3D. Μπορούμε να συμπεράνουμε λοιπόν εν κατακλείδι ότι η ParametricPlot3D είναι γενικότερη της Plot3D. Π.χ αν θέλουμε να σχεδιάσουμε το σαμάρι δεν έχουμε παρά να θέσουμε την x ίση με την πρώτη παράμετρο u και την y με την v και το z(δηλ. το ύψος f[x,y]) ίση με f[u,v]:

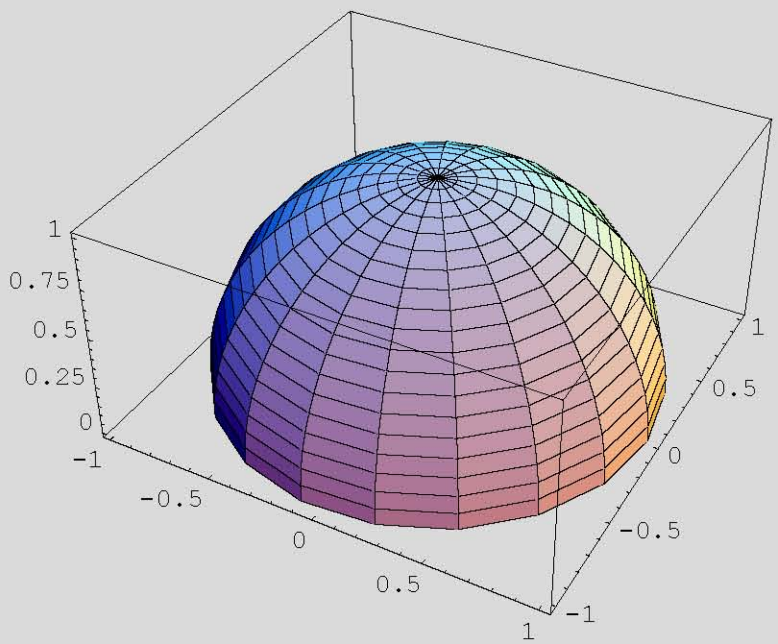
```
ParametricPlot3D[{u, v, u^2 / 9 - v^2 / 4}, {u, -1, 2}, {v, -1, 1}];
```



Ως προς τις επιλογές διαφέρει ελάχιστα απο την Plot3D. Βασικά έχουμε διαφορετικές προεπιλεγμένες τιμές στα PlotPoints, και BoxRatios που είναι PlotPoints→Automatic καιBoxRatios→Automatic.

Με παραμετρικές εξισώσεις μπορούμε να ορίσουμε (και να σχεδιάσουμε κατά συνέπεια)τον τόρο, διάφορες κυλινδρικές επιφάνειες, παραβολοειδή, την σφαίρα κ.ο.κ. Παρακάτω δίνουμε την γραφική παράσταση ενός κομματιού της μοναδιαίας σφαίρας(με ακτινα ίση με 1) και με κέντρο την αρχή των αξόνων. Κατα'ρχην γράφουμε βέβαια την σφαίρα παραμετρικά με την χρήση των σφαιρικών συντεταγμένων!

```
ParametricPlot3D[
  {Cos[theta] Sin[phi], Sin[theta] Sin[phi], Cos[phi]},
  {theta, 0 , 2 Pi}, {phi, 0, Pi / 2}];
```

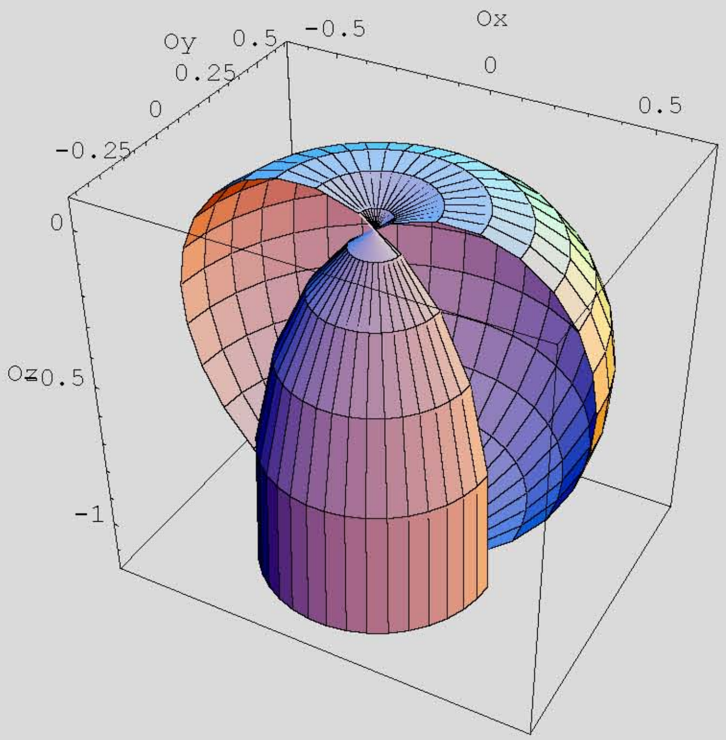


Άσκηση. Για να δούμε πως σχηματίζεται η παραπάνω επιφάνεια καθώς το phi αυξάνει θα μπορούσατε να δοκιμάζατε το Animate στο πακέτο <<Graphics`Animation`. Μελετήστε τις δυνατότητες του. Επίσης θα μπορούσατε να τρέξετε και το παρακάτω πρόγραμμα(πατήστεδιπλό κλίκ σε ένα απ' αυτά):

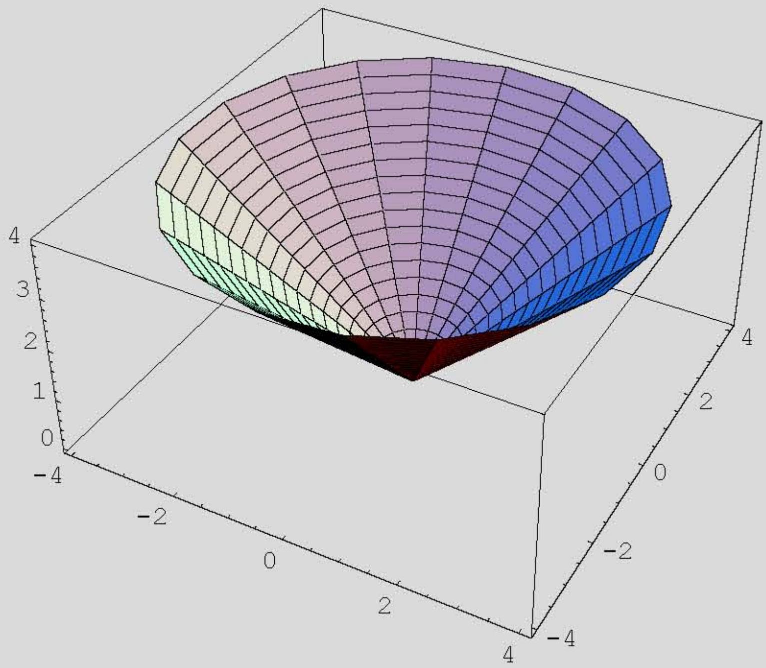
```
Table[ParametricPlot3D[{Cos[theta] Sin[phi],
    Sin[theta] Sin[phi], Cos[phi]}, {theta, 0 , 2 Pi},
   {phi, 0, n Pi / 2}, PlotRange → {{-1, 1}, {-1, 1}, {0, 1}},
   Ticks → None], {n, 0.1, 1, 0.05}];
```

3. Ειδιαίτερα στις περιπτώσεις που η επιφάνεια έχει εξίσωση r=f[theta,phi](όπου r είναι η απόσταση του σημείου απο την αρχή των αξόνων) **σε σφαιρικές συντεταγμένες**, θα ήταν προτιμότερο να χρησιμοποιήσουμε την SphericalPlot3D. Όμοια, όταν η επιφάνεια μας ορίζεται με **κυλινδρικές συντεταγμένες** με την εξίσωση z=g[r,theta] όπου (r,theta,z) οι κυλινδρικές συντεταγμένες, τότε θα ήταν προτιμότερο να χρησιμοποιήσουμε την CylindricalPlot3D. Οι κυλινδρικές συντεταγμένες είναι η γενίκευση των πολικών του επιπέδου. Επίσης πρέπει να ξέρουμε ότι το r παίρνει μη αρνητικές τιμές ενώ η theta απο 0 έως 2 Pi και η phi απο 0 έως Pi. Το z φυσικά μπορεί να πάρει οποιαδήποτε πραγματική τιμή. Σε κάθε όμως περίπτωση πρέπει να καλέσουμε το πακέτο Graphics`ParametricPlot3D`. Π.χ

```
<< Graphics`ParametricPlot3D`
SphericalPlot3D[Log[phi], {phi, Pi / 10 , Pi},
  {theta, 0, Pi}, AxesLabel → {"Ox", "Oy", "Oz"}];
```

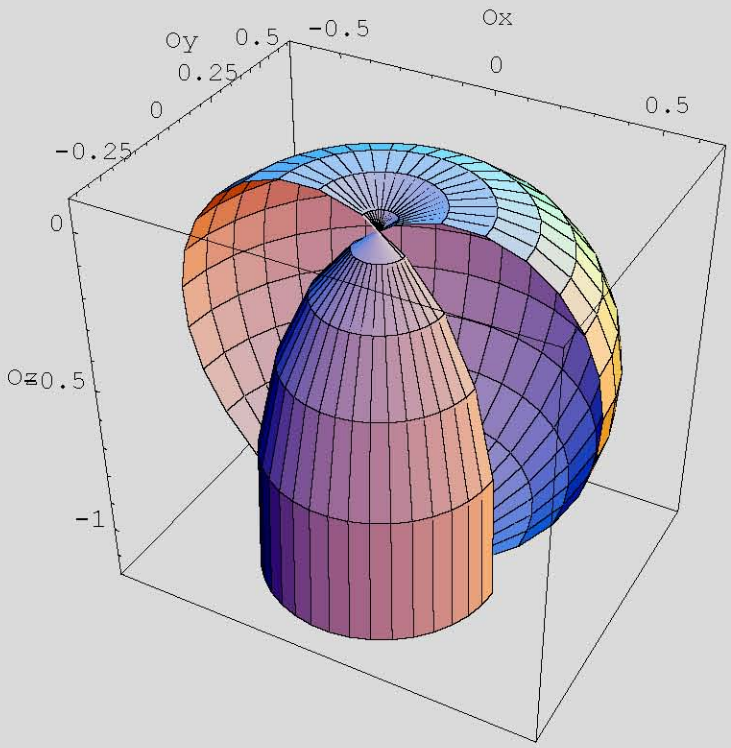


```
CylindricalPlot3D[r, {r, 0, 4}, {theta, 0, 2 Pi}];
```



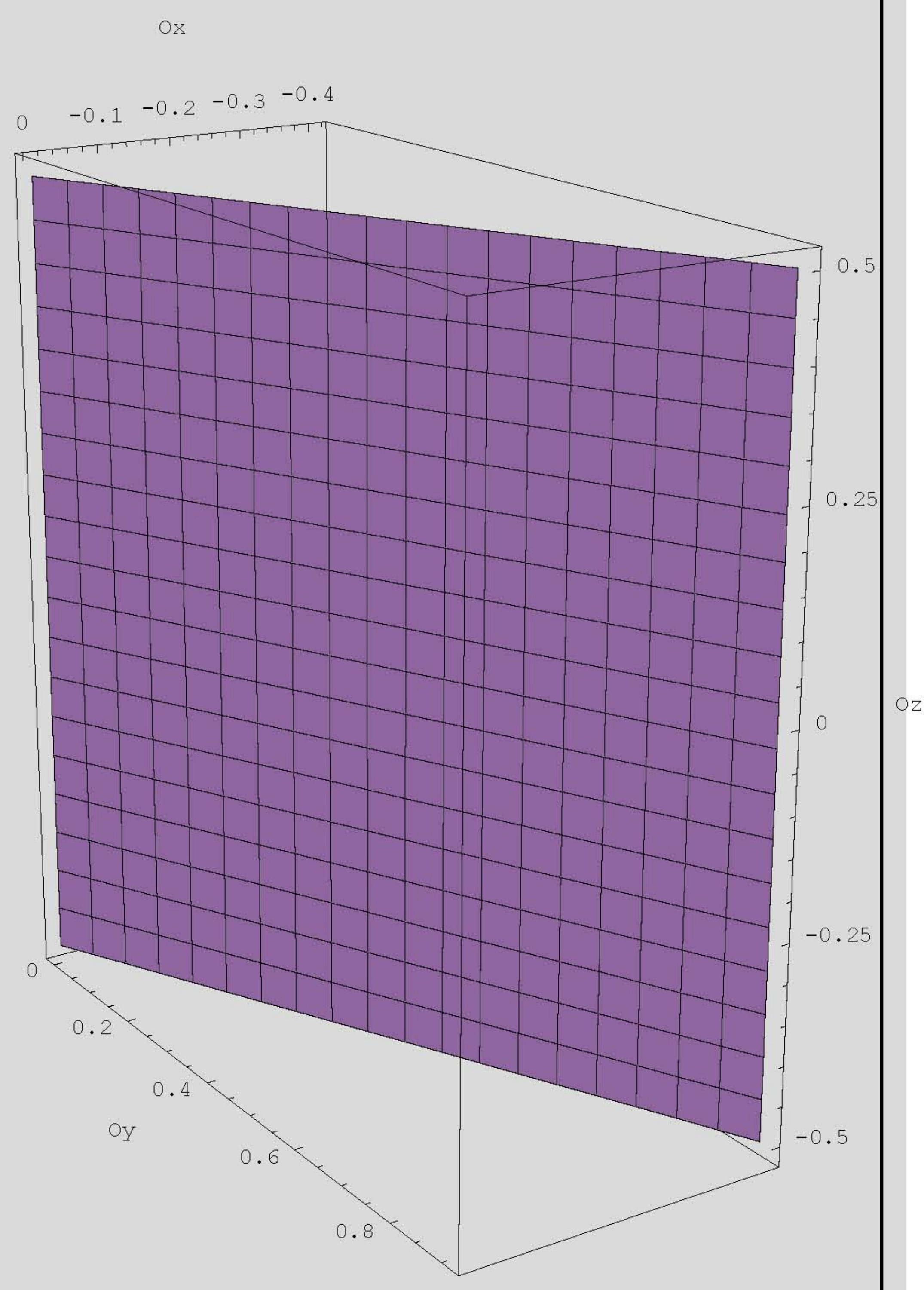
Μπορούμε να σχεδιάσουμε ξανά τις παραπάνω επιφάνειες χρησιμοποιώντας αποκλειστικά την ParametricPlot3D. Θα πρέπει όμως να γνωρίζουμε τους τύπους μετασχηματισμών μεταξύ καρτεσιανών και πολικών ή κυλινδρικών συντεταγμένων . Ετσι για παράδειγμα η Log[phi] σε καρτεσινές είναι x= Log[phi] Sin[phi] Cos[theta], y=Log[phi] Sin[phi] Sin[theta],z=Log[phi] Cos[phi] οπότε:

```
ParametricPlot3D[{Log[phi] Sin[phi] Cos[theta],
   Log[phi] Sin[phi] Sin[theta], Log[phi] Cos[phi]},
  {phi, Pi/10 , Pi}, {theta, 0, Pi}, AxesLabel → {"Ox", "Oy", "Oz"}];
```



Παρατήρηση: Αν οι κυλινδρικές ή οι σφαιρικές επιφάνειες μας **δεν** δίνονται απο συναρτήσεις της μορφής που περιγράψαμε παραπάνω, τότε δεν μπορούμε να χρησιμοποιήσουμε τις CylindricalPlot3D και SphericalPlot3D ή θα πρέπει να λύσουμε ως προς z ή r αντίστοιχα! Για παράδειγμα, να σχεδιάσουμε την επιφάνεια με εξίσωση theta=2 (δηλ. 2 rad) σε κυλινδρικές συντεταγμένες. Σε τέτοιες περιπτώσεις θα πρέπει να χρησιμοποιήουμε την γνωστή μας ParametricPlot3D! Π.χ

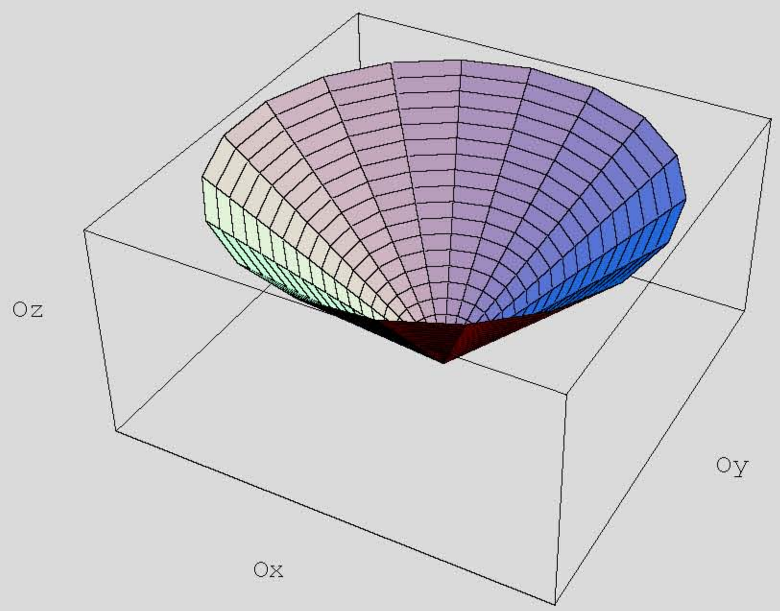
```
ParametricPlot3D[{r Cos[2], r Sin[2], z},
  {r, 0, 1}, {z, -.5, .5}, AxesLabel → {"Ox", "Oy", "Oz"},
  AxesEdge → {{-1, 1}, {1, -1}, {-1, 1}},
  ViewPoint -> {1.743, 2.690, 1.085}, ImageSize → 400];
```



Πρόκειται λοιπόν για επίπεδο! Βάλαμε AxesEdge→{{-1,1},{1,-1},{-1,1}}για να εμφανιστούν τα Ticks στις συγκεκριμμένες πλευρές του κουτιού.

4. Ένας κύλινδρος είναι μια επιφάνεια η οποία έχει κάποιο άξονα, γύρω απο τον οποίο είναι τοποθετημένη συμμετρικά. Ορίζουμε τις **εκ περιστροφής επιφάνειες** εκείνες τις επιφάνειες που προκύπτουν απο την περιστροφή μιας επίπεδης καμπύλης γύρω απο μια ευθεία που λέγεται **άξονας**. Χωρίς περιορισμό της γενικότητας, μπορούμε να υποθέσουμε ότι ο άξονας περιστροφής είναι ο Oz και η καμπύλη βρίσκεται πάνω στο Oxz επίπεδο και δίνεται απο την εξίσωση z=f[x]. Τότε με την εντολή SurfaceOfRevolution[f[x],{x,xmin,xmax}] παίρνουμε την ζητούμενη επιφάνεια εκ περιστροφής.Για να εκτελεστεί η εντολή SurfaceOfRevolution θα πρέπει πρώτα να καλέσουμε το υποπρόγραμμα SurfaceOfRevolution του πακέτου Graphics. Π.χ για να σχεδιάσουμε ένα κώνο ύψους ίσο με 1 θα πρέπει να περιστρέψουμε την z=x δηλ. την διαγώνιο του Oxz επιπεδου γύρω απο τον Oz:

```
<< Graphics`SurfaceOfRevolution`
SurfaceOfRevolution[x, {x, 0, 1},
  Ticks → None, AxesLabel → {"Ox", "Oy", "Oz"}];
```

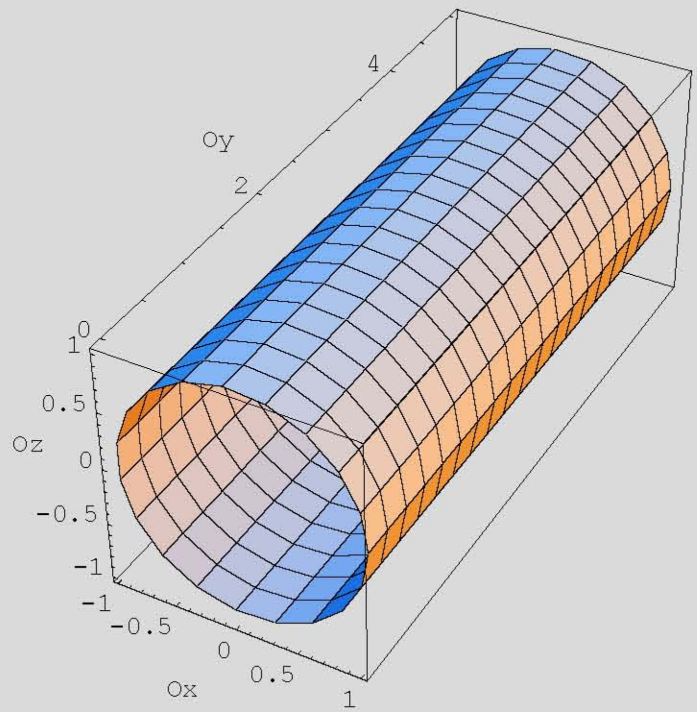


Για να σχεδιάσουμε ένα απλο κύλινδρο γύρω απο τον Oz θα πρέπει να περιστρέψουμε την ευθεία x= κάποια σταθερά, γύρω απο τον Oz. Επειδή η καμπύλη με εξίσωση x=σταθερά **δεν** ειναι συνάρτηση του x θα χρειαστεί να χρησιμοποιήσουμε μια γενικοτερη μορφή της εντολής SurfaceOfRevolution. Ο γενικότερος τύπος είναι ο

SurfaceOfRevolution[{f[t],g[t],h[t]},{t,tmin,tmax},RevolutionAxis->{a,b,c}]

όπου τώρα η καμπύλη δίνεται παραμετρικά στο χώρο με τον τύπο {x,y,z}= {f[t],g[t],h[t]} όπου t είναι η παράμετρος. Επίσης με RevolutionAxis->{a,b,c} θέτουμε άξονα περιστροφής την ευθεία που διέρχεται απο την αρχή των αξόνων (0,0,0) και το σημείο του χώρου με συντεταγμένες {a,b,c}. Π.χ για να σχεδιάσουμε τον κύλινδρο με ακτίνα ίση με 1 και μήκους 5 που περιστρέφεται γύρω απο τον άξονα Oy θα γράψουμε

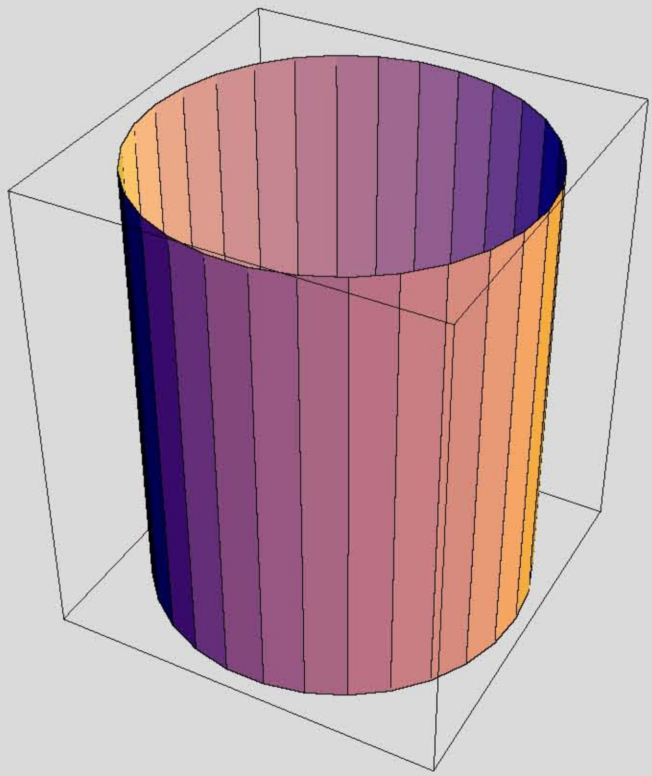
```
SurfaceOfRevolution[{1, t, 0}, {t, 0, 5},
  RevolutionAxis → {0, 1, 0}, AxesLabel → {"Ox", "Oy", "Oz"}];
```



Η {1,t,0} για 0<=t<=5 είναι η ευθεία που διέρχεται απο τα σημεία (1,0,0) και (1,5,0). Προσπαθείστε να την εντοπίσετε...Ο άξονας περιστροφής είναι η ευθεία που διέρχεται απο τα (0,0,0) και (0,1,0) δηλ. είναι ο άξονας Oy. Παρακάτω θα δούμε πως να σχεδιάζουμε τον άξονα περιστροφής και την καμπύλη περιστροφής μαζί με το γράφημα.

5. Με την εντολή Show[Graphics3D[Σχήμα1[a,b,...],Graphics3D[Σχήμα2[ ]],...] μπορούμε να πάρουμε άμμεσα και χωρίς πολύπλοκες πληκτρολογήσεις τη γραφική παράσταση ενός η περισσότερων γνωστών Σχήμα(των). Οι σταθερές a,b,...είναι παράμετροι, οι τιμές των οποίων επηρεάζουν τη γραφική παράσταση. Για να δείτε ακριβώς πως, ζητείστε σχετική βοήθεια απο το Help. Το απαραίτητο πακέτο είναι το Graphics`Shapes`. Το Σχήμα μπορεί να είναι έλικα, διπλή έλικα, τόρο, κύλινδρος, κώνος, σφαίρα, λωρίδα του Mobius(MoebiusStrip), πολύγωνο, τετράγωνο κουτί(Cuboid), πολύεδρα(τετράεδρα, κ.ο.κ), ευθύγραμμο τμήμα (Line),σημείο, κείμενο(Text) κ.ο.κ Π.χ για να σχεδιάσουμε ένα κύλινδρο ύψους 1.2 και ακτίνας 0.5 και για την σχεδίαση να χρησιμοποιηθούν 30 πολύγωνα θα γράψουμε:

```
<< Graphics`Shapes`
Show[Graphics3D[Cylinder[0.5, 0.6, 30]]];
```

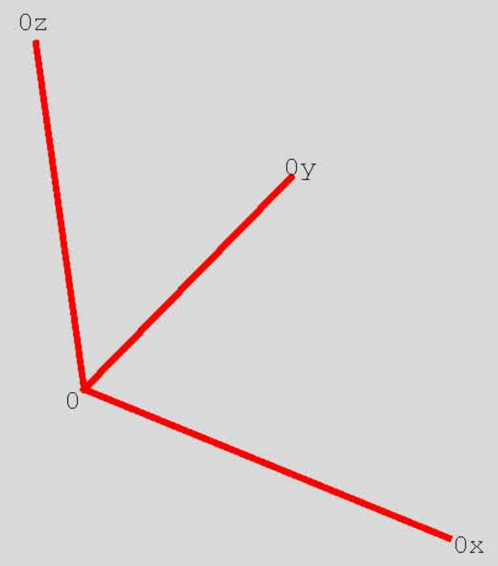


Προσέξτε ότι **δεν** εμφανίζονται άξονες, αυτό διότι η προεπιλεγμένη τιμή για το Graphics3D είναι Axes→False. Με χρήση των κατάλληλων εντολών μπορούμε να θέσουμε και το ορθογώνιο καρτεσιανό σύστημα αξόνων στο γράφημά μας. Για το σκοπό αυτό ορίσαμε το tmhma [x1_,y1_,z1_,x2_,y2_,z2_] που είναι ένα κόκκινο ευθύγραμμο τμήμα που ενώνει τα σημεία (x1,y1,z1) και (x2,y2,z2) και βαλαμε και ετικέτες (στο σημείο x,y,z) με την χρήση της etiketa.

```
etiketa[text_, x_, y_, z_] := Graphics3D[Text[text, {x, y, z}]]
```

```
tmhma[x1_, y1_, z1_, x2_, y2_, z2_] :=
 Graphics3D[{RGBColor[1, 0, 0], Thickness[.008],
   Line[{{x1, y1, z1}, {x2, y2, z2}}]}]
```

```
systhma = {tmhma[0, 0, 0, 1, 0, 0],
   etiketa["0x", 1.05, 0, 0], tmhma[0, 0, 0, 0, 1, 0],
   etiketa["0y", 0, 1.05, 0] , tmhma[0, 0, 0, 0, 0, 1],
   etiketa["0z", 0, 0, 1.05], etiketa["0", 0, -0.05, 0]};
Show[systhma, Boxed → False];
```



Άσκηση. Σχεδιάστε τον άξονα περιστροφής και την καμπύλη περιστροφής μαζί με τον κύλινδρο εκ περιστροφής που είδαμε παραπάνω δηλ. τον SurfaceOfRevolution[{1,t,0},{t,0,5},RevolutionAxis→{0,1,0}, AxesLabel→{"Ox","Oy","Oz"}].

9.3.2 Άλλες δυνατότητες των τρισδιάστατων γραφικών.

Μπορούμε να κάνουμε Animations με πολλους τρόπους. Το πακέτο <<Graphics`Animation` είναι αρκετά χρήσιμο και αξίζει να το μελετήσετε. Παρακάτω δίνουμε μια απλή περιστροφή ενός σχήματος που έχουμε ξαναδεί με την χρήση ης SpinShow.

```
<< Graphics`Animation`
<< Graphics`ParametricPlot3D`
grafhma = Show[systhma,
   SphericalPlot3D[Log[phi], {phi, Pi / 10 , Pi}, {theta, 0, Pi},
    DisplayFunction → Identity], DisplayFunction → Identity];
SpinShow[grafhma, Frames → 20, SpinRange → {0 Degree, 360 Degree},
  SpinDistance → 3, Boxed → False];
```

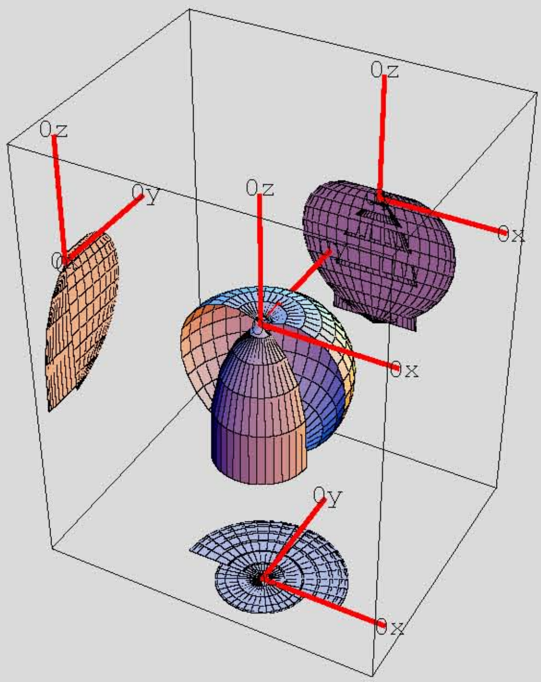
Άσκηση. Με την SpinShow ο άξονας περιστροφής του γραφήματος συνέχεια μετακινείται. Για να αντιμετωπίσετε το πρόβλημα αυτό μπορείτε να χρησιμοποιήσετε την Rotate κατάλληλα. Προσπαθείστε να καταλάβετε τι κάναμε στο παρακάτω πρόγραμμα και πειραματειστείτε με τις εντολές.

```
Table[Show[RotateShape[Graphics3D[MoebiusStrip[]], n Pi / 2, 0, 0],
   PlotRange → {{-2, 2}, {-2, 2}, {-2, 2}}], {n, 0.1, 2, 0.02}];
```

Άσκηση: Προσπαθείστε να βρείτε ένα τρόπο επίδειξης έτσι ώστε όχι μόνο να περιστρέφεται η επιφάνεια, αλλά και να πλησιάζουμε όλο και περισσότερο πάνω σε αυτην...

Με την εντολή Shadow[καποιο γράφημα] μπορούμε να πάρουμε προβολές της γραφικής παράστασης μιας επιφάνειας πάνω στα τρία (ή σε κάποια απ' αυτά) επίπεδα των ορθογωνίων επιπέδων. Πρέπει όμως πρώτα να καλέσουμε το κατάλληλο πακέτο π.χ

```
<< Graphics`Graphics3D`
Shadow[grafhma, ImageSize → 200]
```



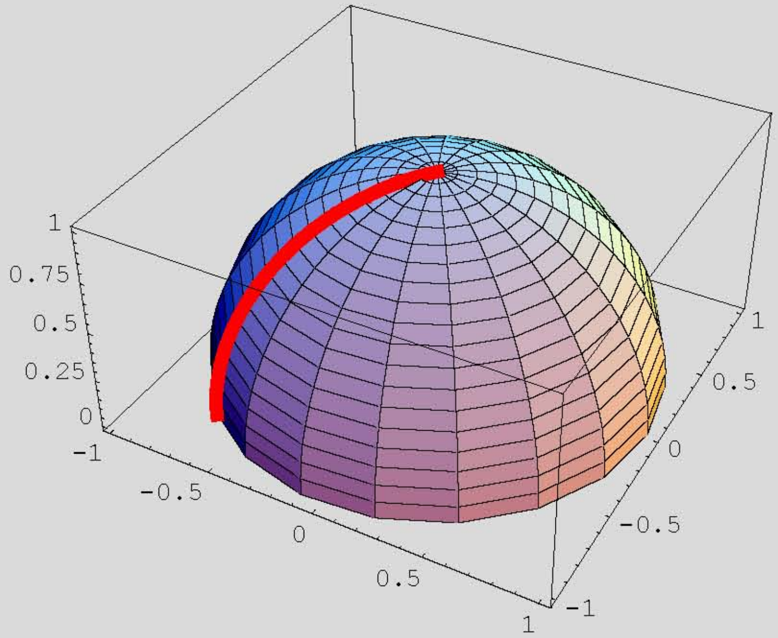
- Graphics3D -

Με ParametricPlot3D[{f_x, f_y, f_z}, {*t*, *tmin*, *tmax*}] μπορούμε να παράγουμε μια τρισδιάστατη καμπύλη στο χώρο που εξαρτάται απο μια παράμετρο t που τρέχει μεταξύ *tmin* και *tmax*. Με αυτό το τρόπο μπορούμε να σχεδιάσουμε κάποιες συγκεκριμμένες καμπύλες πάνω σε μια επιφάνεια π.χ

```
epifaneia = ParametricPlot3D[
   {Cos[theta] Sin[phi], Sin[theta] Sin[phi], Cos[phi]},
   {theta, 0 , 2 Pi}, {phi, 0, Pi / 2}, DisplayFunction → Identity];
```

```
kampylh = ParametricPlot3D[{Cos[4] Sin[phi], Sin[4] Sin[phi],
    Cos[phi], {RGBColor[1, 0, 0], Thickness[.018]}},
   {phi, 0, Pi / 2}, DisplayFunction → Identity];
```

```
Show[kampylh, epifaneia, DisplayFunction → $DisplayFunction];
```



Προσέξτε ότι δεν υπάρχει το PlotStyle όπως το είχαμε δει σε επίπεδα γραφήματα! Οι οδηγίες για το χρώμα και τα λοιπά πρέπει να μπαίνουν μέσα στον ορισμό της καμπύλης!

9.3.3 Γενικές Παρατηρήσεις

Θα πρέπει να ξέρουμε ότι οι τρισδιάστατες γραφικές παραστάσεις έχουν αρκετές διαφορές με σχέση με αυτές που γνωρίσαμε στο επίπεδο. Για παράδειγμα

1)δεν υπάρχει συνάρτηση ImplicitPlot3D. Θα πρέπει να την αντικαταστήσουμε με την ContourPlot3D . Την τελευταία την έχουμε δεί όταν μιλούσαμε για ολοκληρώματα και μπορείτε να πάτε στο κεφάλαιο 7 για να δείτε κάποια παραδείγματα χρήσης της. Είναι πολύ χρήσιμη. Δείτε και την άσκηση που ακολουθεί. Με Contours→{v1,v2,..}μπορούμε να σχεδιάσουμε όσες "φέτες" της επφάνειας f[x,y,z]==v1, f[x,y,z]==v2,.. θέλουμε. Για Contours η πρεπιλεγμένη τιμή είναι η .0 δηλαδή θα σχεδιαστεί μόνο η επιφάνεια με εξίσωση f[x,y,z]==0. Προσέξτε ότι πάιζει ρόλο η σειρά που δίνουμε τα πεδία ορισμού των x, y,z. Δείτε και την άσκηση παρακάτω και αλλάξτε την σειρά των παιδίων ορισμού των μεταβλητών για να δείτε το αποτέλεσμα!

2) Είναι αδύνατο με την Plot3D να σχεδιάσουμε μια λίστα απο συναρτήσεις. Σε τέτοιες περιπτώσεις θα πρέπει να σχεδιάσουμε ξεχωριστά κάθε μια επιφάνεια και στο τέλος να χρησιμοποιήσουμε την Show για το τελικό αποτέλεσμα. Η Show μπορεί να χρησιμοποιειθεί ακόμα και άν οι επιφάνειες ή/και οι καμπύλες μας έχουν σχεδιαστεί με διαφορετικό πακέτο...

3)Το AspectRatio θέλει ιδιαίτερη προσοχή όταν το χρησιμοποιούμε σε τρισδιάστατα γραφικά. μην ξεχνάμε ότι έχει σχέση μόνο με το ορθωγώνιο πλάισιο που (αόρατα) περιβάλλει το γράφημα. Άρα δεν έχει σχέση με τους άξονες ή την ίδια την γραφική παράσταση. Τα ίδια ισχύουν και για το Aspectratio-> Automatic.

4)Δυστυχώς δεν υπάρχει εντολή η Arrow για να ζωγραφίζουμε βέλη στο τρισδιάστο χώρο σε αντίθεση με το

επίπεδο...Έτσι αν θέλουμε να ζωγραφίσουμε τους άξονες με βέλη πρέπει να πειραματιστούμε λίγο με την Graphics3D.
5) Δεν υπάρχει η επιλογή PlotStyle για επιφάνειες. Πρέπει να μελετήσετε τα Options για να βρείτε πως θα αλλάξετε τον φωτισμό ή τον χρωματισμό της επιφάνειας.

Άσκηση: Σχεδιάστε την επιφάνεια με εξίσωση z=2x^2+3 y^2 σε όποιο πεδίο ορισμού θέλετε. Λύσετε ώς προς y (με το χέρι, με την Solve ή όπως αλλιώς θέλετε) και σχεδιάστε ξανά την επιφάνεια y=f[x,z] που προκύπτει. Συγκρίνετε τα αποτελέσματα που βλέπετε με την ContourPlot3D[z-2x^2-3 y^2 ,{x,-1,1},{z,-1,1},{y,-1,1}, PlotPoints→{5,5},Axes→True, AxesLabel→{"Ox","Oz","Oy"}].