

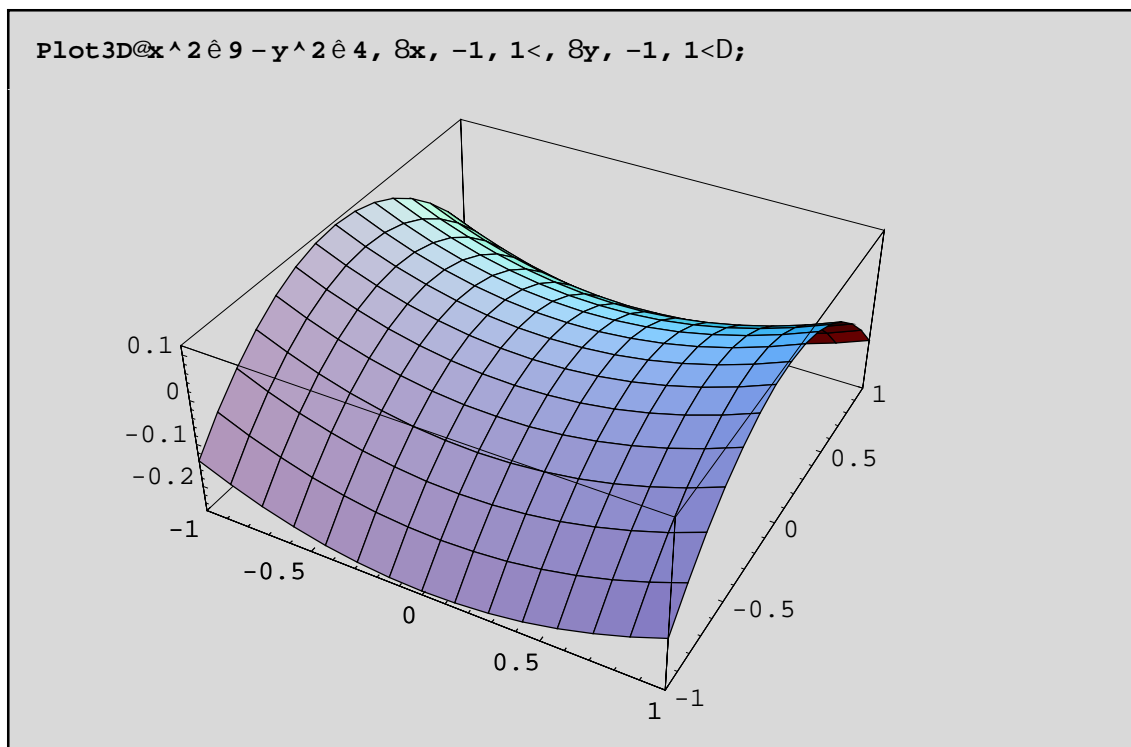
9.3 Τρισδιάστατες Γραφικές Παραστάσεις

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.

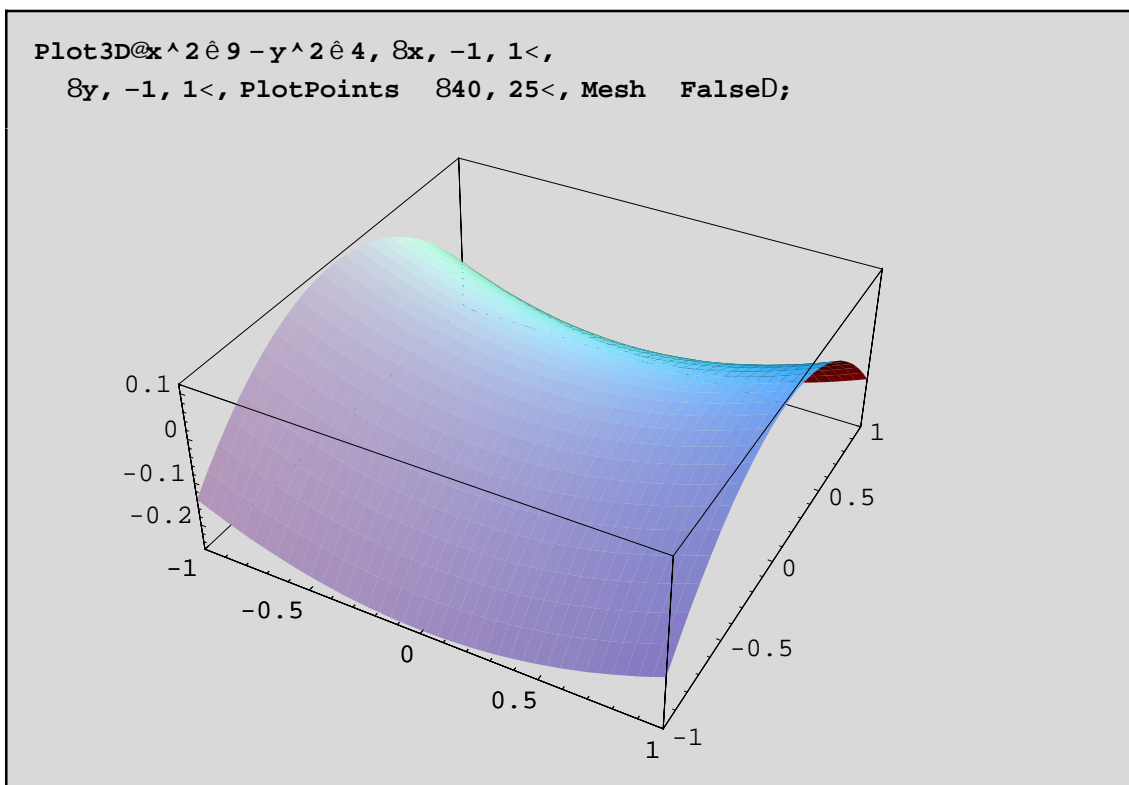
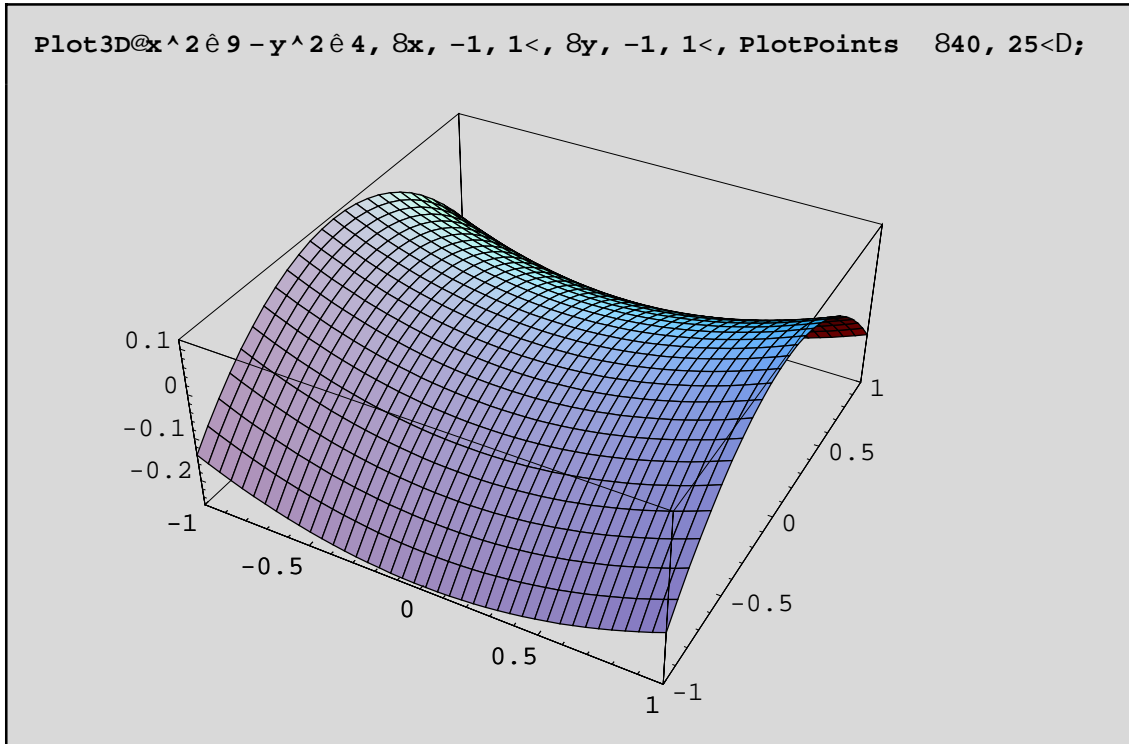
9.3.1 Γραφικές παραστάσεις επιφανειών.

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

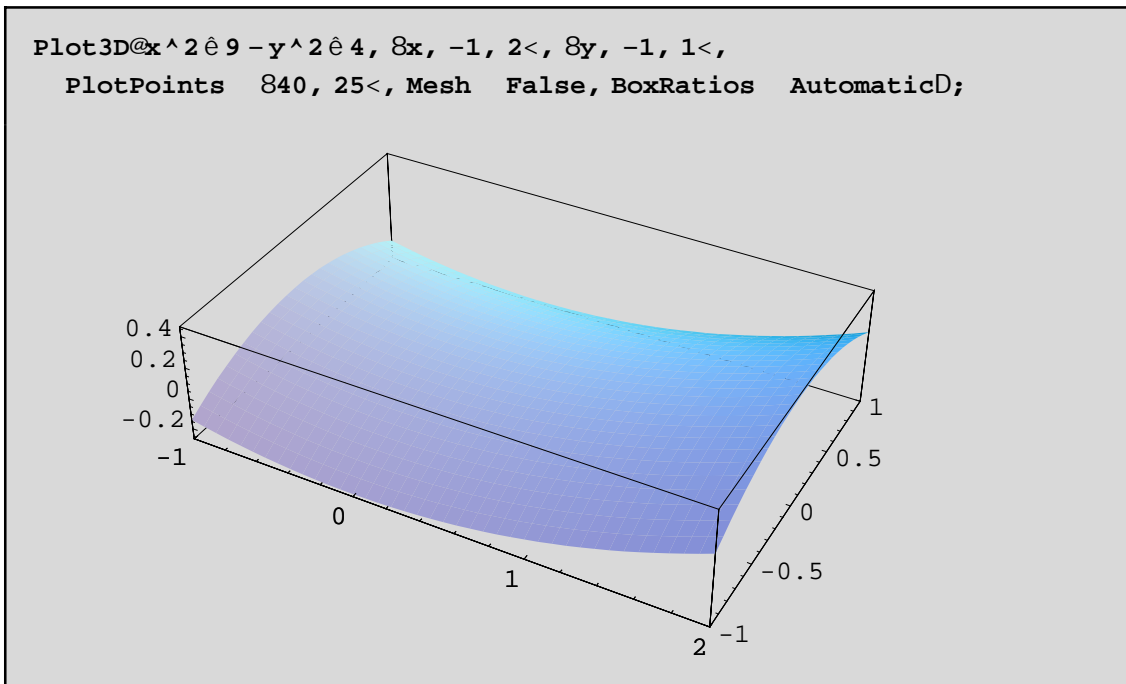
1. Η επιφάνεια είναι το **γράφημα μιας συνάρτησης** δυο μεταβλητών(π.χ της x και y). Τότε χρησιμοποιούμε την `Plot3Dz` π.χ εαν $f(x, y) = 9 - x^2 - 4y^2$ και πεδίο ορισμού το $[-1, 1] \times [-1, 1]$ τότε παίρνουμε την εντολή



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 15×15 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). Με `Mesh->False` σχεδιάζεται η επιφάνεια μας χωρίς να είναι εμφανές το πλέγμα πάνω σ'αυτήν. Π.χ



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. κατά 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



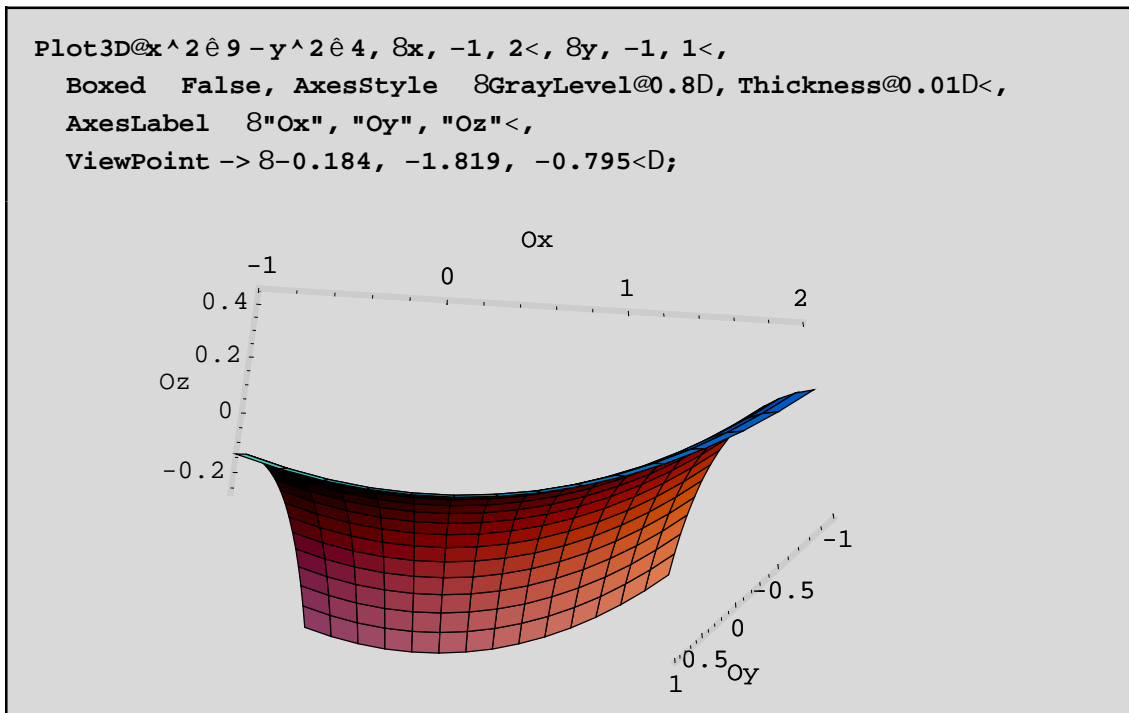
Με `BoxRatios->Automatic` έχουμε ίσες μονάδες μέτρησης σε κάθε άξονα και άρα το "φυσιολογικό" γράφημα της επιφάνειας. Αυτό βέβαια δεν είναι πάντα επιθυμητό γιατί χάνουμε κάποια χαρακτηριστικά της επιφάνειας. Π.χ δεν φαίνεται καθαρά το "σαμαρωτό" σχήμα της επιφάνειας. Με `BoxRatios->{1,1,2.5}` θα έχετε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

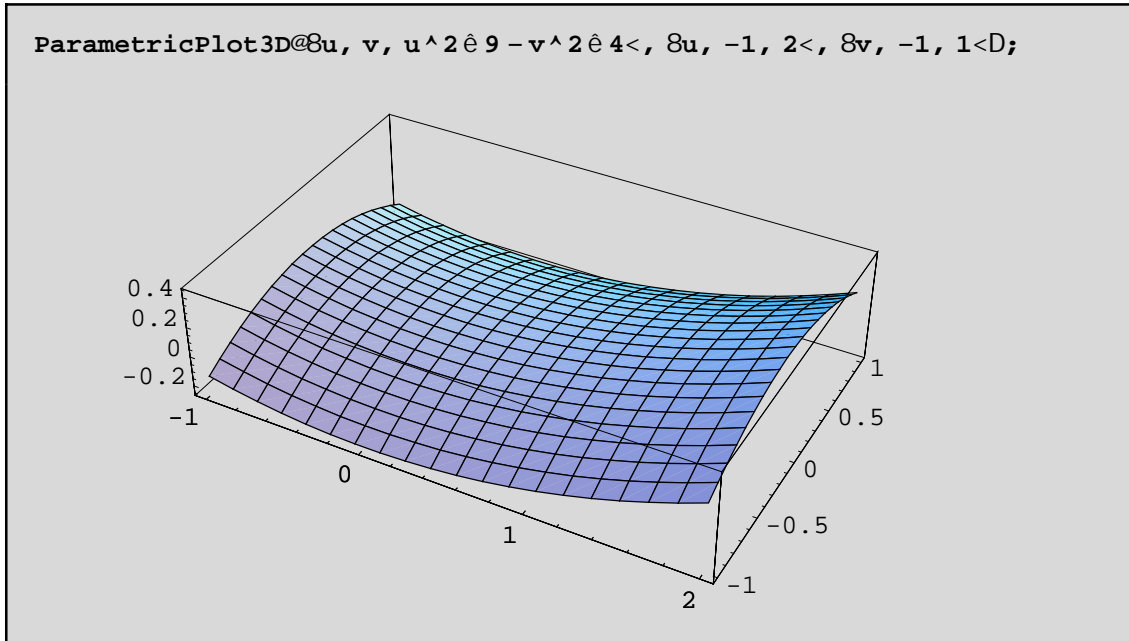
```
Options@ParametricPlot3D
8AmbientLight GrayLevel@0.D, AspectRatio Automatic,
  Axes True, AxesEdge Automatic, AxesLabel None,
  AxesStyle Automatic, Background Automatic, Boxed True,
  BoxRatios Automatic, BoxStyle Automatic, ColorOutput Automatic,
  Compiled True, DefaultColor Automatic, Epilog 8<,
  FaceGrids None, ImageSize Automatic, Lighting True,
  LightSources 8881., 0., 1.<, RGBColor@1, 0, 0D<, 881., 1., 1.<,
  RGBColor@0, 1, 0D<, 880., 1., 1.<, RGBColor@0, 0, 1D<<,
  Plot3Matrix Automatic, PlotLabel None, PlotPoints Automatic,
  PlotRange Automatic, PlotRegion Automatic,
  PolygonIntersections True, Prolog 8<, RenderAll True,
  Shading True, SphericalRegion False, Ticks Automatic,
  ViewCenter Automatic, ViewPoint 81.3, -2.4, 2.<,
  ViewVertical 80., 0., 1.<, DefaultFont f $DefaultFont,
  DisplayFunction f $DisplayFunction,
  FormatType f $FormatType, TextStyle f $TextStyle<
```

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 επικολλήσουμε 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 "στυλ" με το οποίο θα σχεδιαστούν οι τρεις κάθετοι άξονες. Παρακάτω σχεδιάζουμε το σαμάρι, εξαφανίζουμε το κουτί, θέτουμε AxesStyle{GrayLevel[0.8], Thickness[0.01]}, θέτουμε επιγραφές σε κάθε άξονα και παρατηρούμε την επιφάνεια απο κάτω(ViewPoint->{-0.184, -1.819, -0.795}):

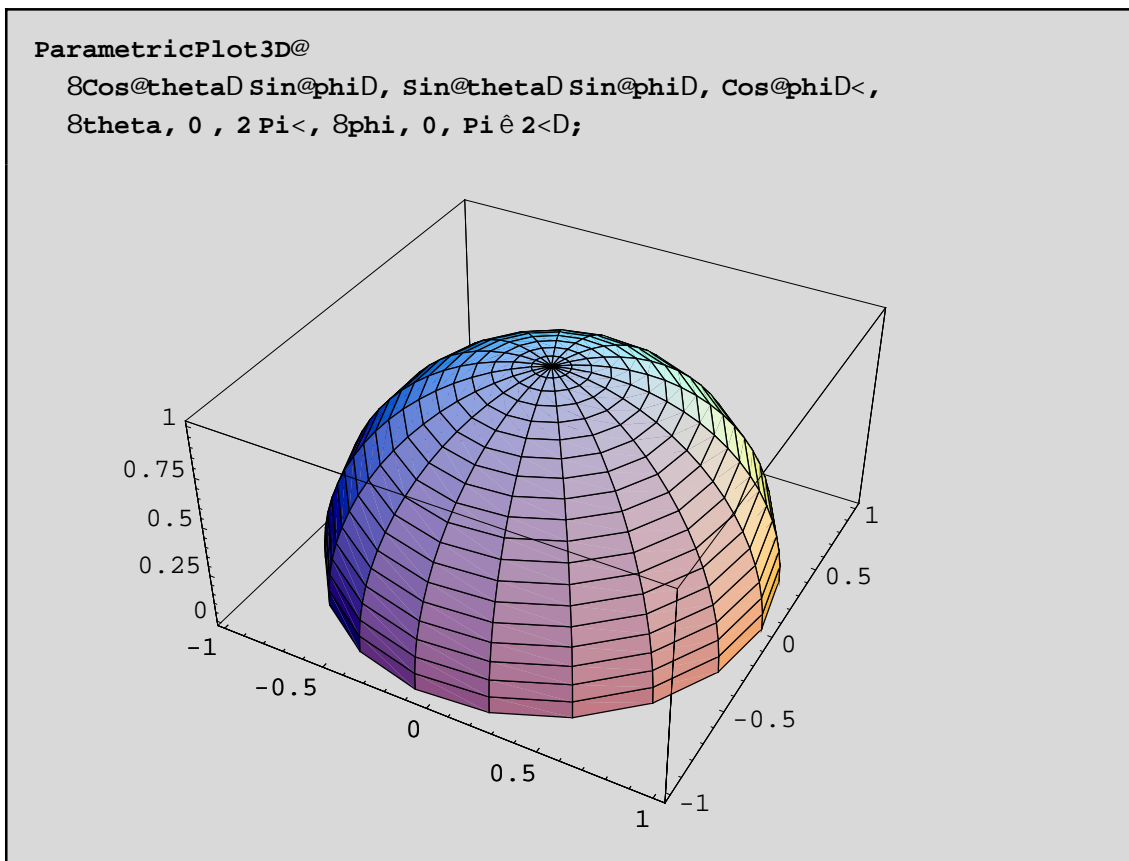


2. Τη γραφική παράσταση μιας επιφάνειας, που ορίζεται με την βοήθεια δύο παραμέτρων (π.χ u και v) την παίρνουμε με την **ParametricPlot3D**. Τέτοιες επιφάνειες λέγονται **παραμετρικές** επιφάνειες. Σε αυτήν την κατηγορία ανήκουν και επιφάνειες που ορίζονται με κυλινδρικές ή σφαιρικές συντεταγμένες. Αυτές 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 $f[u,v]$:

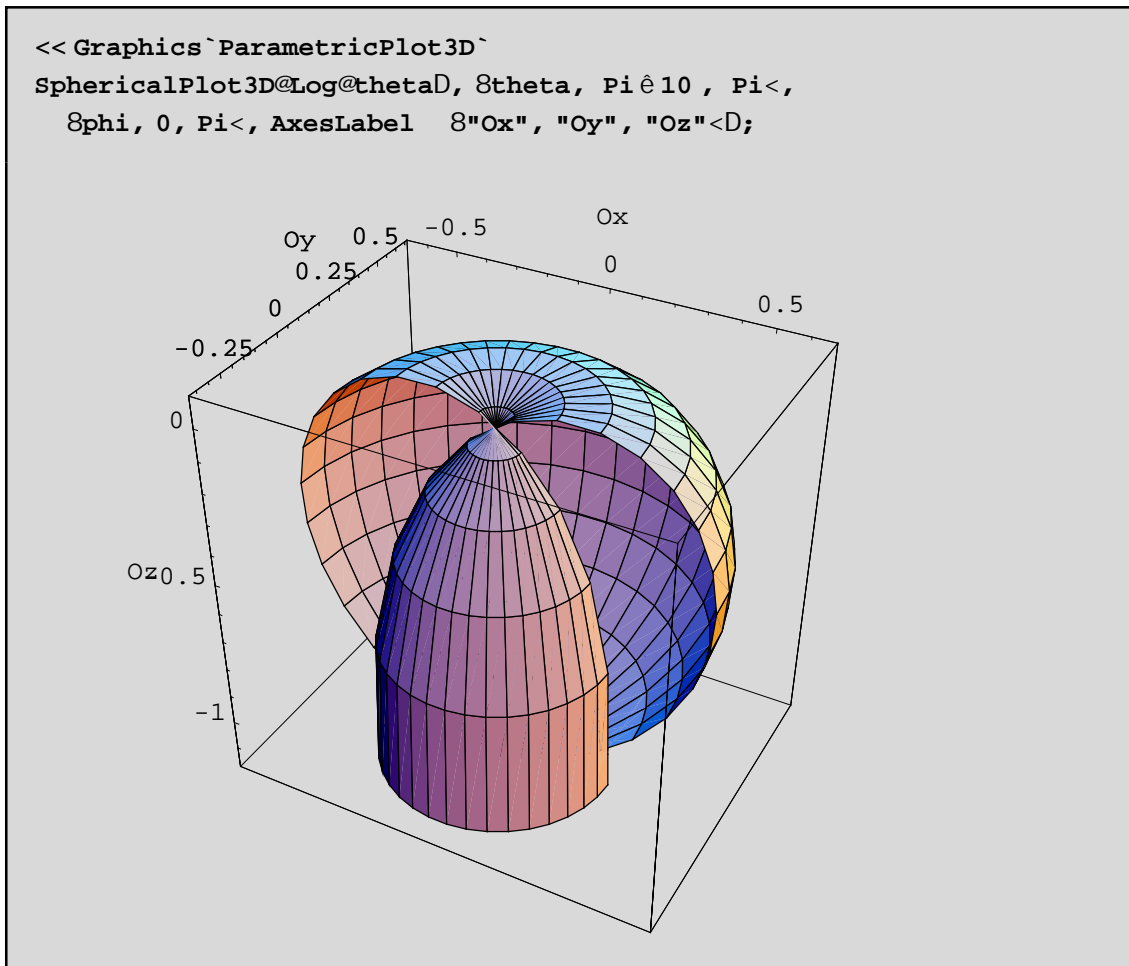


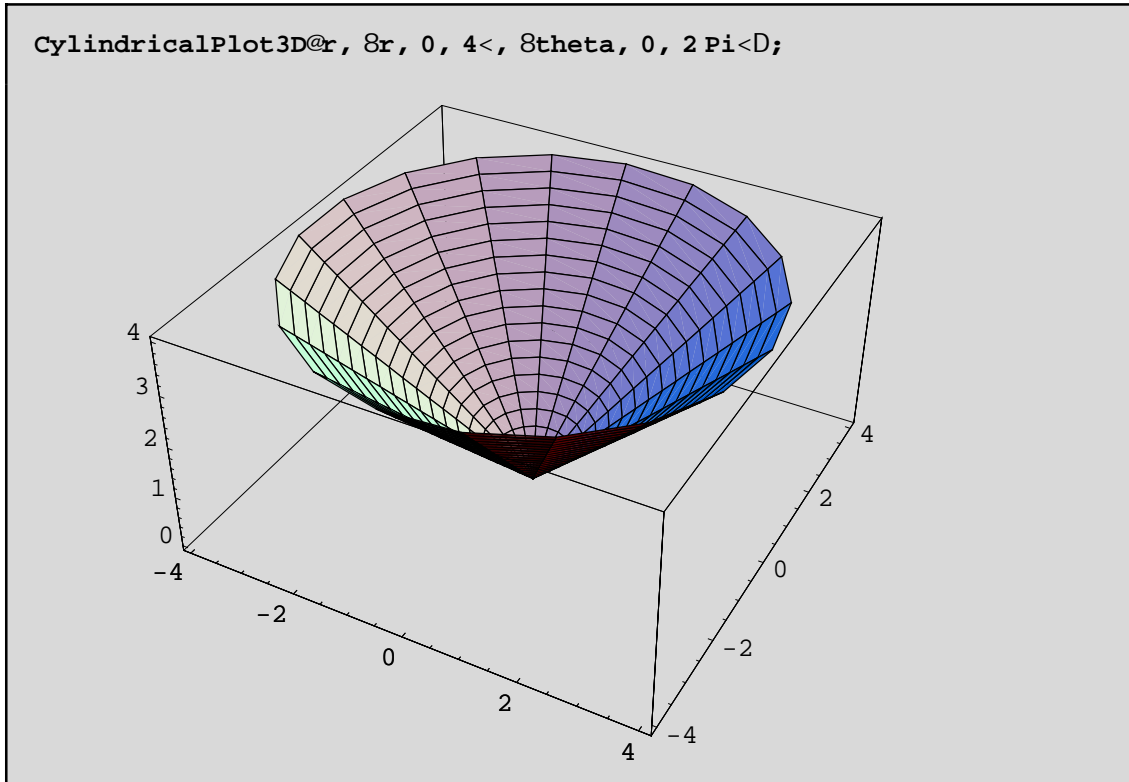
Ως προς τις επιλογές διαφέρει ελάχιστα από την Plot3D. Βασικά έχουμε διαφορετικές προεπιλεγμένες τιμές στα PlotPoints, και BoxRatios που είναι PlotPointsAutomatic και BoxRatiosAutomatic.

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 ακτίνα ίση με 1) και με κέντρο την αρχή των αξόνων. Κατ'ρχην γράφουμε βέβαια την σφαίρα παραμετρικά με την χρήση των σφαιρικών συντεταγμένων!

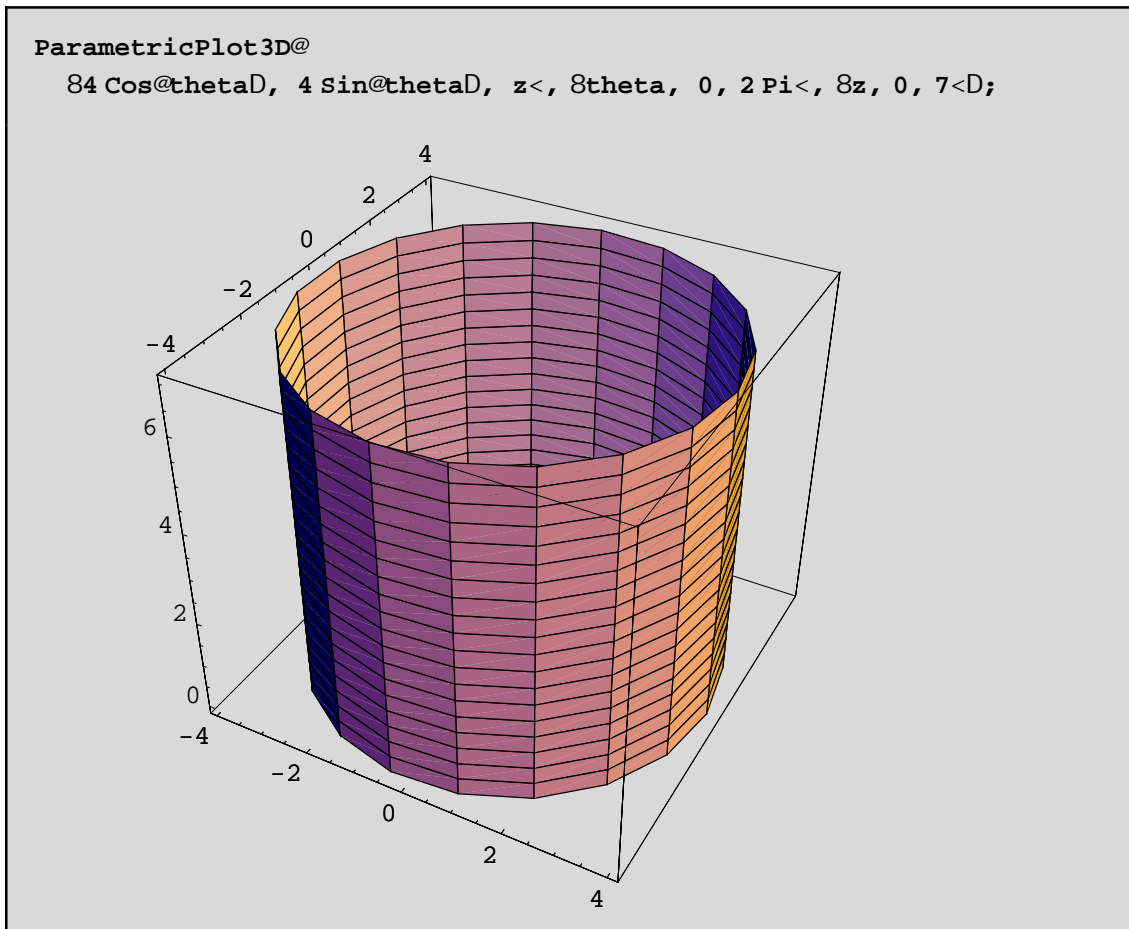


3. Ειδικότερα στις περιπτώσεις που η επιφάνεια έχει εξίσωση $r=f[\theta,\phi]$ (όπου r είναι η απόσταση του σημείου από την αρχή των αξόνων) σε σφαιρικές συντεταγμένες, θα ήταν προτιμότερο να χρησιμοποιήσουμε την `SphericalPlot3D`. Όμοια, όταν η επιφάνεια μας ορίζεται με κυλινδρικές συντεταγμένες με την εξίσωση $z=g[r,\theta]$ όπου (r,θ,z) οι κυλινδρικές συντεταγμένες, τότε θα ήταν προτιμότερο να χρησιμοποιήσουμε την `CylindricalPlot3D`. Σε κάθε όμως περίπτωση πρέπει να καλέσουμε το πακέτο `Graphics`ParametricPlot3D``. Π.χ

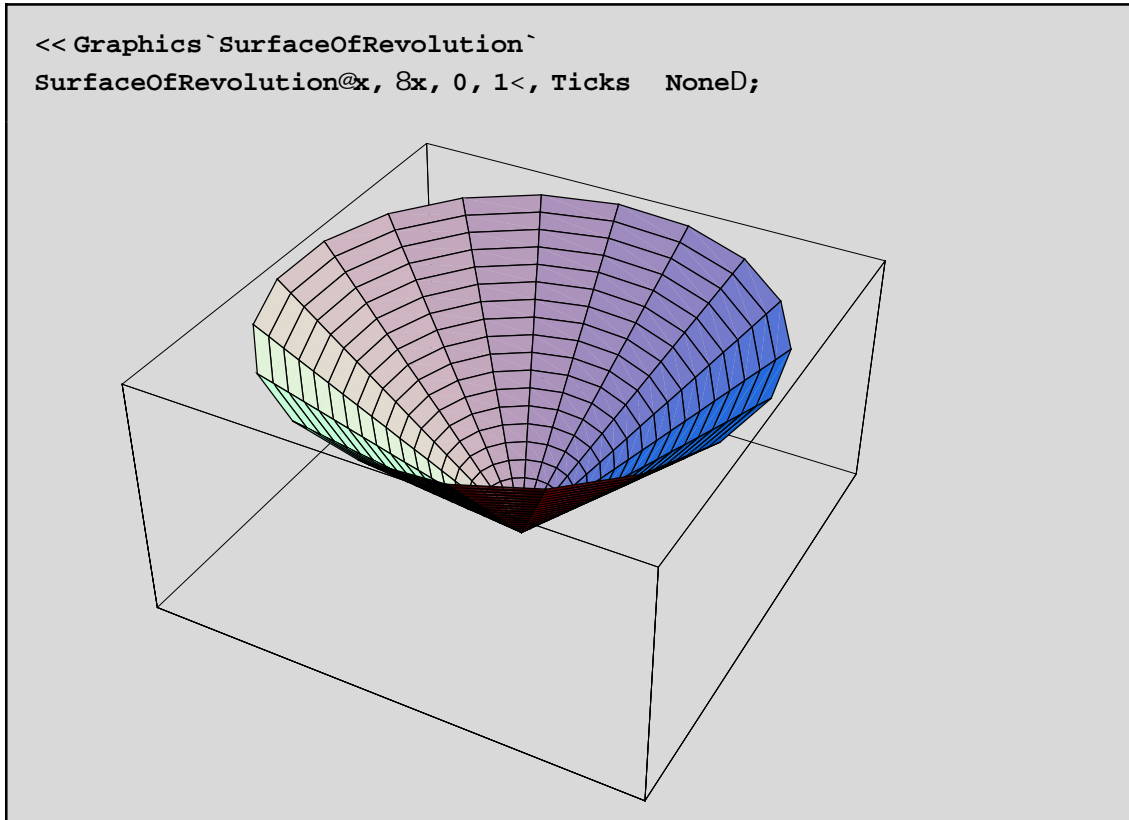




Παρατήρηση: Αν οι κυλινδρικές ή οι σφαιρικές επιφάνειες μας **δεν** δίνονται από συναρτήσεις της μορφής που περιγράψαμε παραπάνω, τότε δεν μπορούμε να χρησιμοποιήσουμε τις `CylindricalPlot3D` και `SphericalPlot3D` ή θα πρέπει να λύσουμε ως προς z ή r αντίστοιχα! Για παράδειγμα, γνωρίζουμε ότι η εξίσωση ενός κυλίνδρου σε κυλινδρικές συντεταγμένες δίνεται με την μορφή $r = \text{μια σταθερά}$ και όχι με την μορφή συνάρτησης των r και θ . Σε τέτοιες περιπτώσεις θα πρέπει να χρησιμοποιήσουμε την γνωστή μας `ParametricPlot3D`! Π.χ



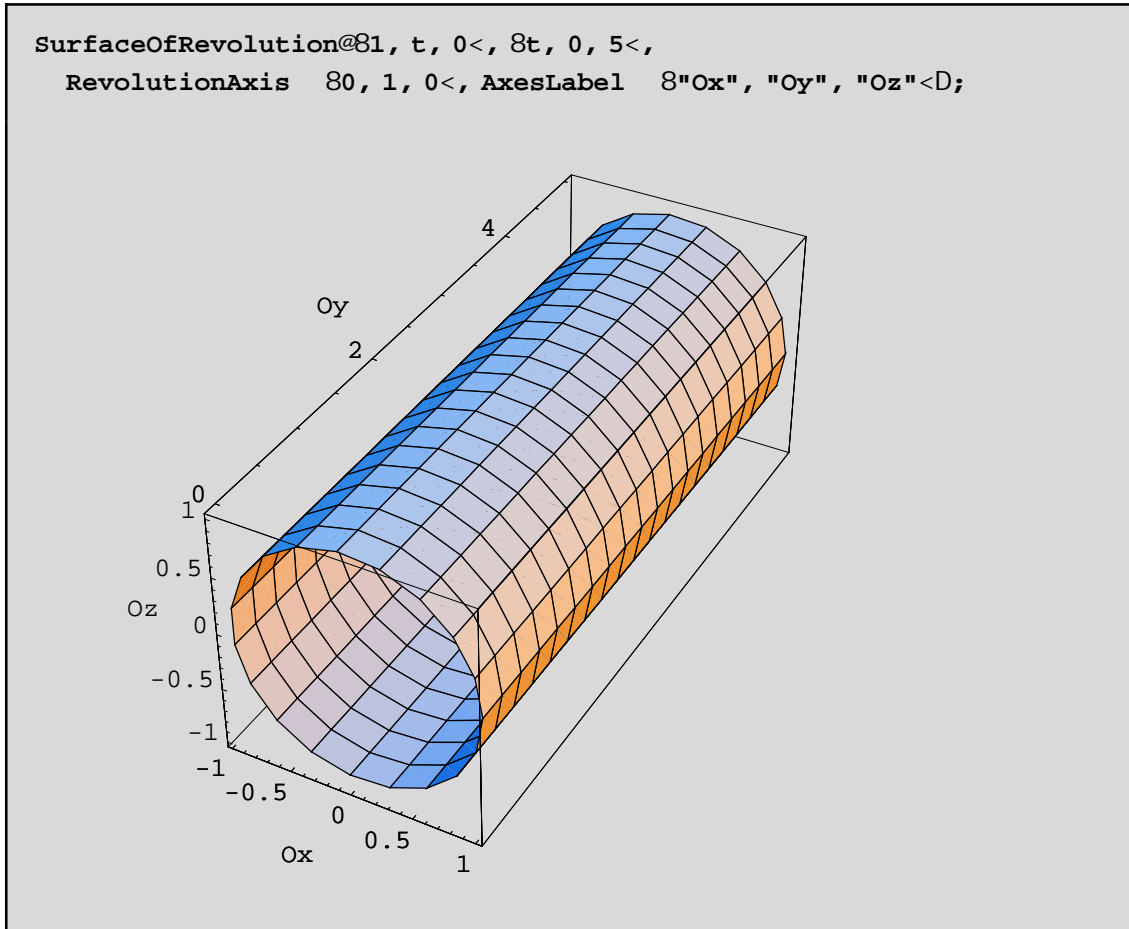
4. Βέβαια, ένας κύλινδρος είναι μια επιφάνεια η οποία έχει κάποιο άξονα, γύρω απο τον οποίο είναι τοποθετημένη συμμετρικά. Ορίζουμε τις **εκ περιστροφής επιφάνειες** εκείνες τις επιφάνειες που προκύπτουν απο την περιστροφή μιας επίπεδης καμπύλης γύρω απο μια ευθεία που λέγεται **άξονας**. Χωρίς περιορισμό 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:



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

$$\text{SurfaceOfRevolution}[\{f[t], g[t], h[t]\}, \{t, t_{\min}, t_{\max}\}, \text{RevolutionAxis} \rightarrow \{a, b, c\}]$$

όπου τώρα η καμπύλη δίνεται παραμετρικά με τον τύπο $\{f[t], g[t], h[t]\}$ όπου t είναι η παράμετρος. Επίσης με RevolutionAxis $\rightarrow \{a, b, c\}$ θέτουμε άξονα περιστροφής την ευθεία που διέρχεται από την αρχή των αξόνων 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 από τον άξονα Oy θα γράψουμε



5. Τελος θα αναφέρουμε ότι με την εντολή `Show[Graphics3D[Σχήμα1[a,b,...],Σχήμα2[],...]]` μπορούμε να πάρουμε άμεσα και χωρίς πολύπλοκες πληκτρολογήσεις τη γραφική παράσταση κάποιων γνωστων Σχήμα(των). Οι σταθερές a, b, \dots 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, κύλινδρος, κώνος, σφαίρα, λωρίδα του Mobius, πολύγωνα, ευθείες 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, 30DDD;
```

