1. Εισαγωγή στο Sage.

1.1 Το μαθηματικό λογισμικό Sage

Το Sage (System for Algebra and Geometry Experimentation) είναι ένα ελεύθερο (δωρεάν) λογισμικό μαθηματικών ανοιχτού κώδικα που υποστηρίζει αριθμητικούς υπολογισμούς, και γενικά την έρευνα και τη διδασκαλία στην άλγεβρα, στη γεωμετρία, στην θεωρία αριθμών, στην κρυπτογραφία, και σε συναφείς τομείς. Καλείται συχνά και Sagemath καθώς η λέξη Sage είναι πολύ κοινή.

Συνδυάζει τις δυνατότητες πολλών υπαρχόντων πακέτων ανοιχτού κώδικα (NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R κ.τ.λ.) σε μία κοινή διεπαφή βασισμένη στη γλώσσα Python. Είναι λογισμικό γενικής χρήσης, με δυνατότητες αναλυτικών και αριθμητικών υπολογισμών καθώς και γραφικών.

Ο γενικός στόχος του Sage, σύμφωνα με τον δημιουργό του, είναι να δημιουργηθεί μια βιώσιμη, δωρεάν, ανοιχτού κώδικα εναλλακτική λύση απέναντι στα μαθηματικά λογισμικά: Maple, Mathematica, Magma, και MATLAB.

Η πρώτη δημόσια έκδοση παρουσιάστηκε τον Φεβρουάριο του 2005 ως ελεύθερο λογισμικό. Ο δημιουργός της είναι ο William Stein, καθηγητής μαθηματικών στο University of Washington.

Εμείς, στη συνέχεια, θα ασχοληθούμε με την έκδοση Sage-7.3 που παρουσιάστηκε τον Αύγουστο του 2016.

Περισσότερα για το Sage στην ιστοσελίδα : http://www.sagemath.org/

1.2 Εγκατάσταση του Sage.

Το μαθηματικό λογισμικό Sage μπορεί να εγκατασταθεί τόσο στο Linux όσο και σε Windows με χρήση εικονικών μηχανών.

Σε περιβάλλον Windows η εγκατάσταση μπορεί να γίνει ακολουθώντας τα παρακάτω βήματα:

1. Κατέβασμα και εγκατάσταση λογισμικού VirtualBox (ελεύθερο λογισμικό δημιουργίας εικονικών μηχανών) από:
   https://www.virtualbox.org/wiki/Downloads
2. Κατέβασμα εικονικής μηχανής SAGE από:
   http://www.sagemath.org/download-windows.html
3. Εγκατάσταση της εικονικής μηχανής SAGE στο VirtualBox:
   https://wiki.sagemath.org/SageApplianceInstallation
4. Εκκίνηση εικονικής μηχανής SAGE με τη βοήθεια του VirtualBox.

5. Άνοιγμα σε ένα πρόγραμμα πλοήγησης (FireFox, Chrome) της ιστοσελίδας: http://localhost:8000.

Επίσης μπορούμε να χρησιμοποιήσουμε το SAGE online, χωρίς την απαίτηση εγκατάστασης λογισμικού στον υπολογιστή μας, μέσω της ιστοσελίδας:

https://cloud.sagemath.com.

Στην περίπτωση αυτή απαιτείται η δημιουργία ενός απλού λογαριασμού ή η σύνδεση μέσω λογαριασμών που έχουμε στο Facebook, ή στο Google, ή στο Twitter.

1.3 Βασικά χαρακτηριστικά του Sage.

α) Οι εντολές στο Sage είναι συναρτήσεις ενσωματωμένες στο πυρήνα (kernel) του Sage και για αυτό το λόγο είναι πάντα διαθέσιμες όταν δουλεύουμε το Sage.

β) Το Sage κάνει διάκριση μεταξύ πεζών και κεφαλαίων.

Για παράδειγμα το Sage αναγνωρίζει και δίνει απάντηση στην εντολή-συνάρτηση:

solve(2*x + 5 ==0, x)

ενώ δεν αναγνωρίζει και οπότε δεν δίνει απάντηση στην εντολή-συνάρτηση:

Solve(2*x + 5 ==0, x)

γιατί θεωρεί ότι τα ονόματα **Solve** και **solve** είναι διαφορετικά, αφού το ένα ξεκινάει με κεφαλαίο γράμμα και το άλλο με πεζό.

Τα ονόματα των προκαθορισμένων εντολών-συναρτήσεων, των μεταβλητών, των επιλογών και των σταθερών που είναι ενσωματωμένες στον πυρήνα του Sage , είναι γραμμένα με πεζά γράμματα. Αν το όνομα αποτελείται από δύο ή περισσότερες λέξεις, οι λέξεις είναι γραμμένες με πεζά γράμματα και χωρίζονται με το σύμβολο της κάτω παύλας «_», για παράδειγμα: ***find_root, find_local_maximum***, κ.τ.λ.

γ) Για τα βασικά μαθηματικά σύμβολα, χρησιμοποιούμε τον εξής συμβολισμό στο Sage:

Μαθηματικός συμβολισμός	Συμβολισμός Sage
π	***pi***
∞ (άπειρο)	***infinity ή Infinity ή oo***
e	***e***
i (μιγαδική μονάδα)	***I ή i***

δ) Τα πάντα στο Sage είναι αντικείμενα. Ότι γράφουμε στο Sage, π.χ. ο ορισμός μίας μεταβλητής με τιμή, ο ορισμός μιας συνάρτηση, μία εξίσωση κ.τ.λ. είναι αντικείμενα. Για παράδειγμα όταν ορίσουμε τη συνάρτηση f = sin(x) δημιουργείται ένα αντικείμενο κλάσης συνάρτησης. Από εδώ και πέρα μπορούμε να δουλέψουμε με την f με πολλούς τρόπους με το να γράψουμε ***f.μέθοδος_για_συνάρτησεις()***, π.χ. για να πάρουμε την παράγωγο της f αρκεί να γράψουμε ***f.diff()***.

```
In [13]: f=sin(x)

In [14]: f.diff()
Out[14]: cos(x)

In [15]: sin(x).diff()
Out[15]: cos(x)
```

1.4 Η Βοήθεια (Help) του Sage

Το Sage έχει μία μεγάλη ενσωματωμένη τεκμηρίωση για τις εντολές-συναρτήσεις και τις σταθερές που χρησιμοποιεί. Πληκτρολογώντας το όνομα μιας συνάρτησης ή μιας σταθεράς, ακολουθούμενη από ένα ερωτηματικό μας δίνει μια συνοπτική περιγραφή και αρκετές φορές παραδείγματα χρήσης της εντολής ή της σταθεράς. Για παράδειγμα:

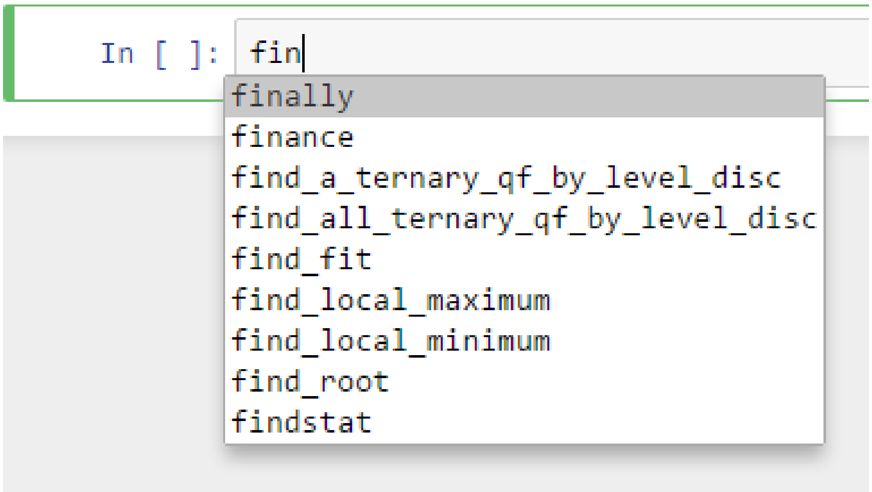
```
In [1]: sin?

In [ ]:
```

```
Type:           Function_sin
String form:    sin
File:           ~/sage-7.3/local/lib/python2.7/site-packages/sage/functions/trig.py
Docstring:
   The sine function.

   EXAMPLES:

      sage: sin(0)
      0
      sage: sin(x).subs(x==0)
      0
      sage: sin(2).n(100)
      0.90929742682568169539601986591
      sage: loads(dumps(sin))
      sin
```

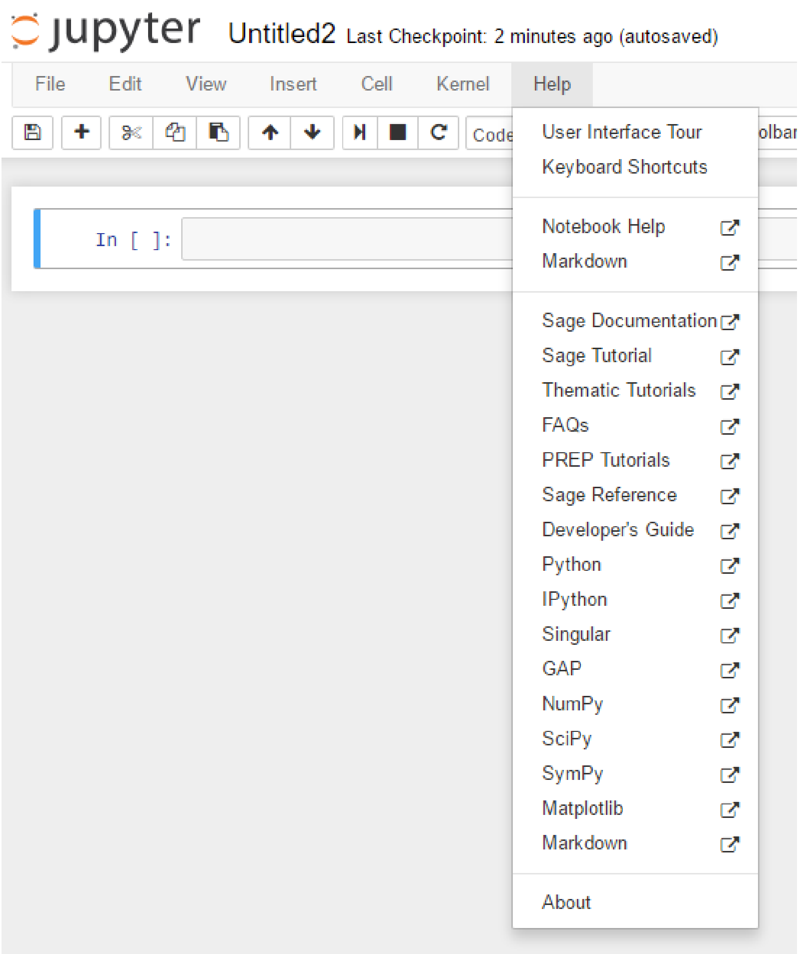
```
In [3]: e?

In [ ]:
```

```
Type:           E
String form:    e
Length:         1
File:           ~/sage-7.3/local/lib/python2.7/site-packages/sage/symbolic/constants_c.pyx
Docstring:
   Dummy class to represent base of the natural logarithm.

   The base of the natural logarithm "e" is not a constant in
   GiNaC/Sage. It is represented by "exp(1)".

   This class provides a dummy object that behaves well under
   addition, multiplication, etc. and on exponentiation calls the
   function "exp".

   EXAMPLES:

   The constant defined at the top level is just "exp(1)":

      sage: e.operator()
      exp
      sage: e.operands()
      [1]

   Arithmetic works:
```

Επίσης γράφοντας τους πρώτους χαρακτήρες μιας εντολής-συνάρτησης και πιέζοντας το πλήκτρο TAB παίρνουμε ένα μενού με όλες τις εντολές-συναρτήσεις που αρχίζουν από αυτούς τους χαρακτήρες.



Τέλος μπορούμε να χρησιμοποιήσουμε τη βοήθεια από το μενού:



Μπορείτε να επισκεφθείτε το Sage Documentation, το Sage Tutorial και το Thematic Tutorial για να γνωρίσετε καλύτερα τις δυνατότητες του Sage.

1.5 Ο πυρήνας (kernel) και η εμπροσθοφυλακή (Front End) του Sage

Το Sage αποτελείται από δύο μέρη. Το πρώτο είναι το παράθυρο εργασίας (Εμπροσθοφυλακή – Front End), το οποίο αποτελεί και τη διεπαφή του χρήστη με το Sage. Το δεύτερο είναι ο πυρήνας (kernel), ο οποίος είναι το μέρος εκείνο του Sage, το οποίο αναλαμβάνει να κάνει όλους τους υπολογισμούς που ζητάει ο χρήστης.

Ο χρήστης δίνει τις εντολές του στο παράθυρο εργασίας (Εμπροσθοφυλακή - Front End), στη συνέχεια οι εντολές μεταφέρονται στο πυρήνα (kernel), ο οποίος κάνει όλους τους απαραίτητους υπολογισμούς και επιστρέφει στο παράθυρο εργασίας το αποτέλεσμα των εντολών που έδωσε αρχικά ο χρήστης.

Για να εκτελεστεί μία εντολή στο Sage πατάμε ταυτόχρονα το πλήκτρο Shift και το πλήκτρο Enter.

Εκτέλεση εντολής:	Shift + Enter.

Το Sage τοποθετεί κάθε εντολή, την οποία πληκτρολογεί ο χρήστης στο παράθυρο εργασίας σε ένα ξεχωριστό κελί. Επίσης η απάντηση που δίνει το Mathematica περιέχεται σε ένα άλλο κελί.

Το Sage αριθμεί αυτόματα κάθε εισαγόμενη εντολή του χρήστη (Input), καθώς και την αντίστοιχη εξερχόμενη απάντηση (Output) με έναν αριθμό. Η αρίθμηση αυτή είναι συνεχής, και αρχίζει από την αρχή σε κάθε νέο notebook που ανοίγουμε στο Sage.

In [n]	Εντολή (input) αριθμός n.
Out[n]	Απάντηση (output) αριθμός n.

Το Sage μας δίνει τη δυνατότητα να αναφερθούμε σε κάποιο συγκεκριμένο από τα προηγούμενα αποτελέσματα, χρησιμοποιώντας το «Out[n]» όπου n είναι ο αριθμός του βήματος του αποτελέσματος που μας ενδιαφέρει. Για παράδειγμα:

```
In [1]: 3+5
Out[1]: 8

In [2]: 2 + Out[1]
Out[2]: 10
```

1.6 Βασικές αριθμητικές πράξεις

Το Sage χρησιμοποιεί για την ανάθεση το σύμβολο του «=» ενώ για τη σύγκεριση τα σύμβολα «==» (ισότητα), «<=» (μικρότερο ή ίσο), «> =» (μεγαλύτερο ή ίσο), «<» (μικρότερο) και «>» (μεγαλύτερο).

```
In [9]: a=5; a
Out[9]: 5

In [10]: 2==3
Out[10]: False

In [11]: 2<3
Out[11]: True

In [13]: a==5
Out[13]: True
```

Το Sage υποστηρίζει όλες τις αριθμητικές πράξεις, και μάλιστα με τον γνωστό τρόπο. Έτσι μπορούμε, να προσθέσουμε δύο αριθμούς χρησιμοποιώντας το γνωστό σύμβολο «+», να αφαιρέσουμε δύο αριθμούς χρησιμοποιώντας το γνωστό σύμβολο «–», να πολλαπλασιάσουμε δύο αριθμούς χρησιμοποιώντας το σύμβολο «*» και να διαιρέσουμε δύο αριθμούς χρησιμοποιώντας το σύμβολο «/».

```
In [1]: 2+3
Out[1]: 5

In [2]: 2-3
Out[2]: -1

In [3]: 2*3
Out[3]: 6

In [4]: 9/3
Out[4]: 3
```

Το σύμβολο «//» μας δίνει το πηλίκο της ακέραιας διαίρεσης, ενώ το σύμβολο «%» μας δίνει το υπόλοιπο της ακέραιας διαίρεσης. Τέλος η ύψωση σε δύναμη αναπαρίσταται με το σύμβολο «^», ή με το σύμβολο «**».

```
In [5]: 10//3
Out[5]: 3
```

```
In [6]: 10%3
Out[6]: 1
```

```
In [7]: 2**3
Out[7]: 8
```

```
In [8]: 2^3
Out[8]: 8
```

Πρέπει να σημειωθεί ότι υπάρχει συγκεκριμένη προτεραιότητα στην εκτέλεση των πράξεων. Συγκεκριμένα, 1η προτεραιότητα έχει η πράξη της ύψωσης σε δύναμη (^, **), 2η προτεραιότητα έχουν οι πράξεις του πολλαπλασιασμού (*) και της διαίρεσης (/, //, %) και 3η προτεραιότητα έχουν οι πράξεις της πρόσθεσης (+) και της αφαίρεσης (-). Μπορούμε να παρακάμψουμε όμως αυτή την προτεραιότητα εκτέλεσης των πράξεων χρησιμοποιώντας παρενθέσεις. Όταν χρησιμοποιούμε παρενθέσεις, πρώτα εκτελούνται οι πράξεις μέσα στην παρένθεση, και μετά οι πράξεις έξω από τις παρενθέσεις, πάντα με την προτεραιότητα εκτέλεσης των πράξεων που αναφέραμε πιο πάνω.

```
In [15]: 4*(10//4)+10%4 == 10
Out[15]: True
```

Το Sage προσπαθεί να δίνει την ακριβέστερη απάντηση που μπορεί, σε σχέση πάντα με τον τύπο των αριθμών που εισάγει ο χρήστης. Όταν, λοιπόν, ζητάμε να μας δώσει το πηλίκο δύο ακεραίων, το Sage προσπαθεί να επιστρέψει έναν ακέραιο. Στην περίπτωση που το πηλίκο δεν είναι ακέραιος μας επιστρέφει την αμέσως μετά τον ακέραιο ακριβέστερη απάντηση, που φυσικά είναι ένας ρητός αριθμός. Για παράδειγμα η διαίρεση 10/3, μας δίνει:

```
In [9]: 10/3
Out[9]: 10/3
```

Για να πάρουμε το πηλίκο της διαίρεσης 10/3 ως πραγματικό αριθμό, αρκεί να εισάγουμε τον έναν από τους δύο ακεραίους με μορφή πραγματικού αριθμού:

```
In [10]: 10./3
Out[10]: 3.33333333333333
```

Το ίδιο αποτέλεσμα μπορούμε να το επιτύχουμε με τη χρήση της συνάρτησης ***n***(expr), όπου expr είναι μια παράσταση:

```
In [14]: n(10/3)

Out[14]: 3.33333333333333
```

n(expr)	Επιστρέφει με προσέγγιση την αριθμητική τιμής της παράστασης expr.
n(expr, digits=n)	Επιστρέφει με προσέγγιση την αριθμητική τιμής της παράστασης expr, χρησιμοποιώντας n ψηφία, στα οποία συμπεριλαμβάνονται και τα ψηφία του ακεραίου μέρους της αριθμητικής τιμής.

```
In [16]: n(10/3, digits=20)

Out[16]: 3.3333333333333333333
```

Αντί για τη συνάρτηση ***n*** μπορούμε να χρησιμοποιήσουμε τη συνάρτηση ***N*** ή τη συνάρτηση ***numerical_approx*** με τις ίδιες παραμέτρους με την ***n***.

```
In [19]: N(10/3, digits=20)

Out[19]: 3.3333333333333333333

In [20]: numerical_approx(10/3, digits=20)

Out[20]: 3.3333333333333333333
```

Μπορούμε τις παραστάσεις των οποίων θέλουμε την αριθμητική τιμή να τις χρησιμοποιήσουμε και ως αντικείμενα:

```
In [21]: a=10/3

In [22]: a.numerical_approx()

Out[22]: 3.33333333333333

In [23]: (10/3).numerical_approx()

Out[23]: 3.33333333333333
```

1.7 Επίλυση αλγεβρικών εξισώσεων

Η βασική συνάρτηση επίλυσης αλγεβρικών εξισώσεων είναι η ***solve***:

solve(eqnς, vars)	Επιλύει τις (την) εξισώσεις (-η) eqns ως προς τις (την) μεταβλητές (-η) vars.

Μία εξίσωση στο Sage εισάγεται με διπλό σύμβολο ισότητας ==, αφού το απλό σύμβολο της ισότητας χρησιμοποιείται από το Sage για αναθέσεις.

Οι ρίζες της εξίσωσης x_0, x_1, … εμφανίζονται σε λίστα υπό τη μορφή ισοτήτων "==", ως εξής:

$$[x==x_0, x== x_1, ...].$$

Όλες οι ρίζες της εξίσωσης εμφανίζονται μόνο μία φορά ανεξάρτητα από τη πολλαπλότητά τους. Αν μας ενδιαφέρει και η πολλαπλότητα της ρίζας τότε πρέπει να χρησιμοποιήσουμε τη μέθοδο ***roots( )***.

Αν έχουμε μία εξίσωση που το δεξιό μέλος της είναι 0, μπορούμε να το παραλείψουμε:

```
In [25]: solve(7*x + 3, x)
Out[25]: [x == (-3/7)]
```

Οι συμβολικές μεταβλητές πρέπει να οριστούν κατάλληλα στο SAGE (με εξαίρεση το x το οποίο έχει προκαθοριστεί ως συμβολική μεταβλητή). Ο τρόπος είναι ο εξής:

y = var (‘y’)	για μία συμβολική μεταβλητή
y, z = var(‘y z’)	για περισσότερες από μία συμβολική σειρά

Οπότε, αν έχουμε μία εξίσωση η οποία περιέχει δύο ή περισσότερες συμβολικές μεταβλητές, αρχικά πρέπει να ορίσουμε τις συμβολικές μεταβλητές και στη πρέπει να ορίσουμε υποχρεωτικά ως προς ποια μεταβλητή θέλουμε να τη λύσουμε:

```
In [5]: y, a, b, c, d = var ('y a b c d')

In [6]: solve(a*y + b == c*x + d, x)
Out[6]: [x == (a*y + b - d)/c]
```

Μπορούμε να λύσουμε την εξίσωση ως προς οποιαδήποτε μεταβλητή της:

```
In [7]: solve(a*y + b == c*x + d, y)
Out[7]: [y == (c*x - b + d)/a]

In [8]: solve(a*y + b == c*x + d, a)
Out[8]: [a == (c*x - b + d)/y]

In [9]: solve(a*y + b == c*x + d, b)
Out[9]: [b == c*x - a*y + d]
```

Στην περίπτωση που έχουμε συστήματα εξισώσεων, ο προσδιορισμός eqns είναι μία λίστα της μορφής [eqn1, eqn2, ...] ενώ ο προσδιορισμός vars είναι μία λίστα της μορφής[var1, var2, ...]

```
In [10]: y, z = var('y z')

In [11]: solve([3*x+z==3, 7*x-9*y-2*z==1, x+y+7*z==3], [x,y,z])
Out[11]: [[x == (155/167), y == (94/167), z == (36/167)]]
```

Με τη συνάρτηση ***solve*** έχουμε ακριβής επίλυση μίας εξίσωσης:

```
In [12]: solve(x**2+5*x-3==0,x)
Out[12]: [x == -1/2*sqrt(37) - 5/2, x == 1/2*sqrt(37) - 5/2]
```

Αν θέλουμε τα αποτελέσματα μας να εμφανίζονται σε μία πιο μαθηματική μορφή θα πρέπει να χρησιμοποιήσουμε την εντολή %display typeset ή την pretty_print_default(True).

%display typeset ή pretty_print_default(True)	εμφανίζονται τα αποτελέσματα σε ποιο μαθηματική μορφή
pretty_print_default(False)	εμφανίζονται τα αποτελέσματα σε ποιο απλή μορφή

```
In [13]: %display typeset

In [14]: solve(x**2+5*x-3==0,x)
```

Out[14]:

$$\left[x = -\frac{1}{2}\sqrt{37} - \frac{5}{2}, x = \frac{1}{2}\sqrt{37} - \frac{5}{2}\right]$$

```
In [15]: pretty_print_default(False)

In [16]: solve(x**2+5*x-3==0,x)
Out[16]: [x == -1/2*sqrt(37) - 5/2, x == 1/2*sqrt(37) - 5/2]
```

Όταν έχουμε πολυωνυμικές εξισώσεις και μας ενδιαφέρει η αριθμητική (προσεγγιστική) επίλυση της εξίσωσης μπορούμε να χρησιμοποιήσουμε, τη μέθοδο ***roots( )***. Η μέθοδος ***roots( )*** επιστρέφει εκτός από τις ρίζες και την πολλαπλότητα της κάθε μιας. Επίσης στη μέθοδος ***roots( )*** μπορούμε να καθορίσουμε και τον δακτύλιο των λύσεων.

(eqn).roots(ring = XX)	λύνει αριθμητικά την εξίσωση eqn=0 στο δακτύλιο XX (ZZ ο δακτύλιος των ακεραίων, QQ ο δακτύλιος των ρητών, RR ο δακτύλιος των πραγματικών και CC ο δακτύλιος των μιγαδικών), επιστρέφοντας τις ρίζες μαζί με τη πολλαπλότητά τους.

Προσοχή στη μέθοδο ***roots( )*** χρησιμοποιούμε μόνο το δεξί μέλος μιας εξίσωσης της οποίας το αριστερό ισούται με μηδέν.

```
In [18]: (x**2+5*x-3).roots(ring=RR)
Out[18]: [(-5.54138126514911, 1), (0.541381265149110, 1)]
```

Με τη μέθοδο roots( ) μπορούμε να λύσουμε την ίδια εξίσωση σε διάφορους δακτυλίους λύσεων. Για παράδειγμα η λύση της εξίσωσης $x^6-2x^2-3x+4=0$ στους ακεραίους αριθμούς δίνει μία λύση πολλαπλότητας 1:

```
In [21]: (x**6-2*x**2-3*x+4).roots(ring=ZZ)
Out[21]: [(1, 1)]
```

στους πραγματικούς αριθμούς δίνει 2 λύσεις πολλαπλότητας 1:

```
In [22]: (x**6-2*x**2-3*x+4).roots(ring=RR)
Out[22]: [(1.00000000000000, 1), (1.06916982139911, 1)]
```

και στους μιγαδικούς αριθμούς δίνει 6 λύσεις, όλες πολλαπλότητας 1:

```
In [23]: (x**6-2*x**2-3*x+4).roots(ring=CC)
Out[23]: [(1.00000000000000, 1),
          (1.06916982139911, 1),
          (-1.17674841649108 - 0.630259924714768*I, 1),
          (-1.17674841649108 + 0.630259924714768*I, 1),
          (0.142163505791523 - 1.44197149031700*I, 1),
          (0.142163505791523 + 1.44197149031700*I, 1)]
```

1. Εισαγωγή στο Sage.

1.1 Το μαθηματικό λογισμικό Sage

Το Sage (System for Algebra and Geometry Experimentation) είναι ένα ελεύθερο (δωρεάν) λογισμικό μαθηματικών ανοιχτού κώδικα που υποστηρίζει αριθμητικούς υπολογισμούς, και γενικά την έρευνα και τη διδασκαλία στην άλγεβρα, στη γεωμετρία, στην θεωρία αριθμών, στην κρυπτογραφία, και σε συναφείς τομείς. Καλείται συχνά και Sagemath καθώς η λέξη Sage είναι πολύ κοινή.

Συνδυάζει τις δυνατότητες πολλών υπαρχόντων πακέτων ανοιχτού κώδικα (NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R κ.τ.λ.) σε μία κοινή διεπαφή βασισμένη στη γλώσσα Python. Είναι λογισμικό γενικής χρήσης, με δυνατότητες αναλυτικών και αριθμητικών υπολογισμών καθώς και γραφικών.

Ο γενικός στόχος του Sage, σύμφωνα με τον δημιουργό του, είναι να δημιουργηθεί μια βιώσιμη, δωρεάν, ανοιχτού κώδικα εναλλακτική λύση απέναντι στα μαθηματικά λογισμικά: Maple, Mathematica, Magma, και MATLAB.

Η πρώτη δημόσια έκδοση παρουσιάστηκε τον Φεβρουάριο του 2005 ως ελεύθερο λογισμικό. Ο δημιουργός της είναι ο William Stein, καθηγητής μαθηματικών στο University of Washington.

Εμείς, στη συνέχεια, θα ασχοληθούμε με την έκδοση Sage-7.3 που παρουσιάστηκε τον Αύγουστο του 2016.

Περισσότερα για το Sage στην ιστοσελίδα : http://www.sagemath.org/

1.2 Εγκατάσταση του Sage.

Το μαθηματικό λογισμικό Sage μπορεί να εγκατασταθεί τόσο στο Linux όσο και σε Windows με χρήση εικονικών μηχανών.

Σε περιβάλλον Windows η εγκατάσταση μπορεί να γίνει ακολουθώντας τα παρακάτω βήματα:

1. Κατέβασμα και εγκατάσταση λογισμικού VirtualBox (ελεύθερο λογισμικό δημιουργίας εικονικών μηχανών) από:
   https://www.virtualbox.org/wiki/Downloads
2. Κατέβασμα εικονικής μηχανής SAGE από:
   http://www.sagemath.org/download-windows.html
3. Εγκατάσταση της εικονικής μηχανής SAGE στο VirtualBox:
   https://wiki.sagemath.org/SageApplianceInstallation
4. Εκκίνηση εικονικής μηχανής SAGE με τη βοήθεια του VirtualBox.

5. Άνοιγμα σε ένα πρόγραμμα πλοήγησης (FireFox, Chrome) της ιστοσελίδας: http://localhost:8000.

Επίσης μπορούμε να χρησιμοποιήσουμε το SAGE online, χωρίς την απαίτηση εγκατάστασης λογισμικού στον υπολογιστή μας, μέσω της ιστοσελίδας:

https://cloud.sagemath.com.

Στην περίπτωση αυτή απαιτείται η δημιουργία ενός απλού λογαριασμού ή η σύνδεση μέσω λογαριασμών που έχουμε στο Facebook, ή στο Google, ή στο Twitter.

1.3 Βασικά χαρακτηριστικά του Sage.

α) Οι εντολές στο Sage είναι συναρτήσεις ενσωματωμένες στο πυρήνα (kernel) του Sage και για αυτό το λόγο είναι πάντα διαθέσιμες όταν δουλεύουμε το Sage.

β) Το Sage κάνει διάκριση μεταξύ πεζών και κεφαλαίων.

Για παράδειγμα το Sage αναγνωρίζει και δίνει απάντηση στην εντολή-συνάρτηση:

solve(2*x + 5 ==0, x)

ενώ δεν αναγνωρίζει και οπότε δεν δίνει απάντηση στην εντολή-συνάρτηση:

Solve(2*x + 5 ==0, x)

γιατί θεωρεί ότι τα ονόματα **Solve** και **solve** είναι διαφορετικά, αφού το ένα ξεκινάει με κεφαλαίο γράμμα και το άλλο με πεζό.

Τα ονόματα των προκαθορισμένων εντολών-συναρτήσεων, των μεταβλητών, των επιλογών και των σταθερών που είναι ενσωματωμένες στον πυρήνα του Sage , είναι γραμμένα με πεζά γράμματα. Αν το όνομα αποτελείται από δύο ή περισσότερες λέξεις, οι λέξεις είναι γραμμένες με πεζά γράμματα και χωρίζονται με το σύμβολο της κάτω παύλας «_», για παράδειγμα: ***find_root, find_local_maximum***, κ.τ.λ.

γ) Για τα βασικά μαθηματικά σύμβολα, χρησιμοποιούμε τον εξής συμβολισμό στο Sage:

Μαθηματικός συμβολισμός	Συμβολισμός Sage
π	***pi***
∞ (άπειρο)	***infinity ή Infinity ή oo***
e	***e***
i (μιγαδική μονάδα)	***I ή i***

δ) Τα πάντα στο Sage είναι αντικείμενα. Ότι γράφουμε στο Sage, π.χ. ο ορισμός μίας μεταβλητής με τιμή, ο ορισμός μιας συνάρτηση, μία εξίσωση κ.τ.λ. είναι αντικείμενα. Για παράδειγμα όταν ορίσουμε τη συνάρτηση f = sin(x) δημιουργείται ένα αντικείμενο κλάσης συνάρτησης. Από εδώ και πέρα μπορούμε να δουλέψουμε με την f με πολλούς τρόπους με το να γράψουμε ***f.μέθοδος_για_συνάρτησεις()***, π.χ. για να πάρουμε την παράγωγο της f αρκεί να γράψουμε ***f.diff()***.

```
In [13]: f=sin(x)

In [14]: f.diff()
Out[14]: cos(x)

In [15]: sin(x).diff()
Out[15]: cos(x)
```

1.4 Η Βοήθεια (Help) του Sage

Το Sage έχει μία μεγάλη ενσωματωμένη τεκμηρίωση για τις εντολές-συναρτήσεις και τις σταθερές που χρησιμοποιεί. Πληκτρολογώντας το όνομα μιας συνάρτησης ή μιας σταθεράς, ακολουθούμενη από ένα ερωτηματικό μας δίνει μια συνοπτική περιγραφή και αρκετές φορές παραδείγματα χρήσης της εντολής ή της σταθεράς. Για παράδειγμα:

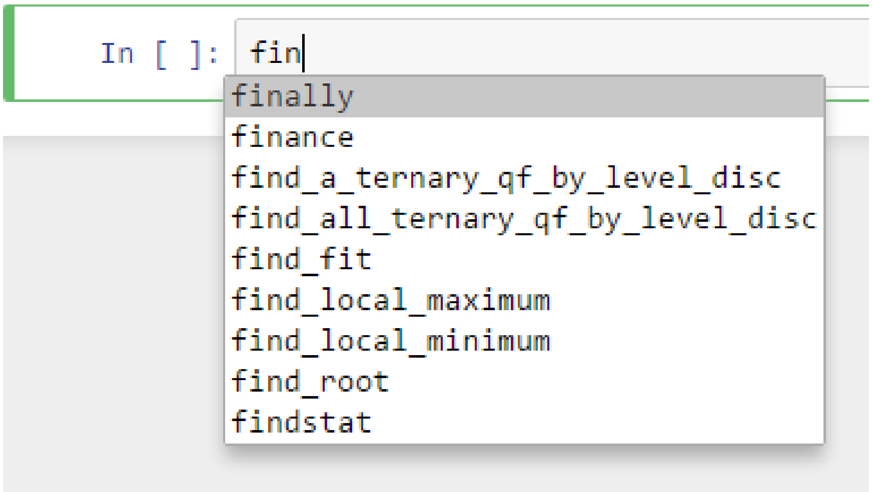
```
In [1]: sin?

In [ ]:
```

```
Type:           Function_sin
String form:    sin
File:           ~/sage-7.3/local/lib/python2.7/site-packages/sage/functions/trig.py
Docstring:
   The sine function.

   EXAMPLES:

      sage: sin(0)
      0
      sage: sin(x).subs(x==0)
      0
      sage: sin(2).n(100)
      0.90929742682568169539601986591
      sage: loads(dumps(sin))
      sin
```

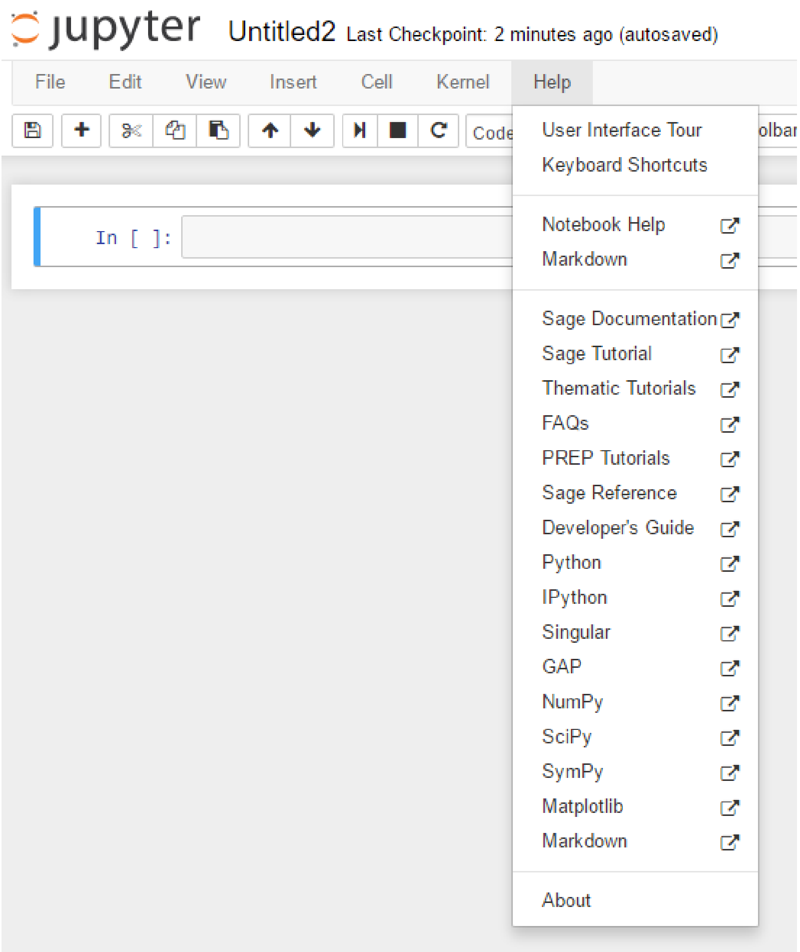
```
In [3]: e?

In [ ]:
```

```
Type:           E
String form:    e
Length:         1
File:           ~/sage-7.3/local/lib/python2.7/site-packages/sage/symbolic/constants_c.pyx
Docstring:
   Dummy class to represent base of the natural logarithm.

   The base of the natural logarithm "e" is not a constant in
   GiNaC/Sage. It is represented by "exp(1)".

   This class provides a dummy object that behaves well under
   addition, multiplication, etc. and on exponentiation calls the
   function "exp".

   EXAMPLES:

   The constant defined at the top level is just "exp(1)":

      sage: e.operator()
      exp
      sage: e.operands()
      [1]

   Arithmetic works:
```

Επίσης γράφοντας τους πρώτους χαρακτήρες μιας εντολής-συνάρτησης και πιέζοντας το πλήκτρο TAB παίρνουμε ένα μενού με όλες τις εντολές-συναρτήσεις που αρχίζουν από αυτούς τους χαρακτήρες.



Τέλος μπορούμε να χρησιμοποιήσουμε τη βοήθεια από το μενού:



Μπορείτε να επισκεφθείτε το Sage Documentation, το Sage Tutorial και το Thematic Tutorial για να γνωρίσετε καλύτερα τις δυνατότητες του Sage.

1.5 Ο πυρήνας (kernel) και η εμπροσθοφυλακή (Front End) του Sage

Το Sage αποτελείται από δύο μέρη. Το πρώτο είναι το παράθυρο εργασίας (Εμπροσθοφυλακή – Front End), το οποίο αποτελεί και τη διεπαφή του χρήστη με το Sage. Το δεύτερο είναι ο πυρήνας (kernel), ο οποίος είναι το μέρος εκείνο του Sage, το οποίο αναλαμβάνει να κάνει όλους τους υπολογισμούς που ζητάει ο χρήστης.

Ο χρήστης δίνει τις εντολές του στο παράθυρο εργασίας (Εμπροσθοφυλακή - Front End), στη συνέχεια οι εντολές μεταφέρονται στο πυρήνα (kernel), ο οποίος κάνει όλους τους απαραίτητους υπολογισμούς και επιστρέφει στο παράθυρο εργασίας το αποτέλεσμα των εντολών που έδωσε αρχικά ο χρήστης.

Για να εκτελεστεί μία εντολή στο Sage πατάμε ταυτόχρονα το πλήκτρο Shift και το πλήκτρο Enter.

Εκτέλεση εντολής:	Shift + Enter.

Το Sage τοποθετεί κάθε εντολή, την οποία πληκτρολογεί ο χρήστης στο παράθυρο εργασίας σε ένα ξεχωριστό κελί. Επίσης η απάντηση που δίνει το Mathematica περιέχεται σε ένα άλλο κελί.

Το Sage αριθμεί αυτόματα κάθε εισαγόμενη εντολή του χρήστη (Input), καθώς και την αντίστοιχη εξερχόμενη απάντηση (Output) με έναν αριθμό. Η αρίθμηση αυτή είναι συνεχής, και αρχίζει από την αρχή σε κάθε νέο notebook που ανοίγουμε στο Sage.

In [n]	Εντολή (input) αριθμός n.
Out[n]	Απάντηση (output) αριθμός n.

Το Sage μας δίνει τη δυνατότητα να αναφερθούμε σε κάποιο συγκεκριμένο από τα προηγούμενα αποτελέσματα, χρησιμοποιώντας το «Out[n]» όπου n είναι ο αριθμός του βήματος του αποτελέσματος που μας ενδιαφέρει. Για παράδειγμα:

```
In [1]: 3+5
Out[1]: 8

In [2]: 2 + Out[1]
Out[2]: 10
```

1.6 Βασικές αριθμητικές πράξεις

Το Sage χρησιμοποιεί για την ανάθεση το σύμβολο του «=» ενώ για τη σύγκεριση τα σύμβολα «==» (ισότητα), «<=» (μικρότερο ή ίσο), «> =» (μεγαλύτερο ή ίσο), «<» (μικρότερο) και «>» (μεγαλύτερο).

```
In [9]: a=5; a
Out[9]: 5

In [10]: 2==3
Out[10]: False

In [11]: 2<3
Out[11]: True

In [13]: a==5
Out[13]: True
```

Το Sage υποστηρίζει όλες τις αριθμητικές πράξεις, και μάλιστα με τον γνωστό τρόπο. Έτσι μπορούμε, να προσθέσουμε δύο αριθμούς χρησιμοποιώντας το γνωστό σύμβολο «+», να αφαιρέσουμε δύο αριθμούς χρησιμοποιώντας το γνωστό σύμβολο «–», να πολλαπλασιάσουμε δύο αριθμούς χρησιμοποιώντας το σύμβολο «*» και να διαιρέσουμε δύο αριθμούς χρησιμοποιώντας το σύμβολο «/».

```
In [1]: 2+3
Out[1]: 5

In [2]: 2-3
Out[2]: -1

In [3]: 2*3
Out[3]: 6

In [4]: 9/3
Out[4]: 3
```

Το σύμβολο «//» μας δίνει το πηλίκο της ακέραιας διαίρεσης, ενώ το σύμβολο «%» μας δίνει το υπόλοιπο της ακέραιας διαίρεσης. Τέλος η ύψωση σε δύναμη αναπαρίσταται με το σύμβολο «^», ή με το σύμβολο «**».

```
In [5]: 10//3
Out[5]: 3
```

```
In [6]: 10%3
Out[6]: 1
```

```
In [7]: 2**3
Out[7]: 8
```

```
In [8]: 2^3
Out[8]: 8
```

Πρέπει να σημειωθεί ότι υπάρχει συγκεκριμένη προτεραιότητα στην εκτέλεση των πράξεων. Συγκεκριμένα, 1η προτεραιότητα έχει η πράξη της ύψωσης σε δύναμη (^, **), 2η προτεραιότητα έχουν οι πράξεις του πολλαπλασιασμού (*) και της διαίρεσης (/, //, %) και 3η προτεραιότητα έχουν οι πράξεις της πρόσθεσης (+) και της αφαίρεσης (-). Μπορούμε να παρακάμψουμε όμως αυτή την προτεραιότητα εκτέλεσης των πράξεων χρησιμοποιώντας παρενθέσεις. Όταν χρησιμοποιούμε παρενθέσεις, πρώτα εκτελούνται οι πράξεις μέσα στην παρένθεση, και μετά οι πράξεις έξω από τις παρενθέσεις, πάντα με την προτεραιότητα εκτέλεσης των πράξεων που αναφέραμε πιο πάνω.

```
In [15]: 4*(10//4)+10%4 == 10
Out[15]: True
```

Το Sage προσπαθεί να δίνει την ακριβέστερη απάντηση που μπορεί, σε σχέση πάντα με τον τύπο των αριθμών που εισάγει ο χρήστης. Όταν, λοιπόν, ζητάμε να μας δώσει το πηλίκο δύο ακεραίων, το Sage προσπαθεί να επιστρέψει έναν ακέραιο. Στην περίπτωση που το πηλίκο δεν είναι ακέραιος μας επιστρέφει την αμέσως μετά τον ακέραιο ακριβέστερη απάντηση, που φυσικά είναι ένας ρητός αριθμός. Για παράδειγμα η διαίρεση 10/3, μας δίνει:

```
In [9]: 10/3
Out[9]: 10/3
```

Για να πάρουμε το πηλίκο της διαίρεσης 10/3 ως πραγματικό αριθμό, αρκεί να εισάγουμε τον έναν από τους δύο ακεραίους με μορφή πραγματικού αριθμού:

```
In [10]: 10./3
Out[10]: 3.33333333333333
```

Το ίδιο αποτέλεσμα μπορούμε να το επιτύχουμε με τη χρήση της συνάρτησης ***n***(expr), όπου expr είναι μια παράσταση:

```
In [14]: n(10/3)
Out[14]: 3.33333333333333
```

n(expr)	Επιστρέφει με προσέγγιση την αριθμητική τιμής της παράστασης expr.
n(expr, digits=n)	Επιστρέφει με προσέγγιση την αριθμητική τιμής της παράστασης expr, χρησιμοποιώντας n ψηφία, στα οποία συμπεριλαμβάνονται και τα ψηφία του ακεραίου μέρους της αριθμητικής τιμής.

```
In [16]: n(10/3, digits=20)
Out[16]: 3.3333333333333333333
```

Αντί για τη συνάρτηση ***n*** μπορούμε να χρησιμοποιήσουμε τη συνάρτηση ***N*** ή τη συνάρτηση ***numerical_approx*** με τις ίδιες παραμέτρους με την ***n***.

```
In [19]: N(10/3, digits=20)
Out[19]: 3.3333333333333333333

In [20]: numerical_approx(10/3, digits=20)
Out[20]: 3.3333333333333333333
```

Μπορούμε τις παραστάσεις των οποίων θέλουμε την αριθμητική τιμή να τις χρησιμοποιήσουμε και ως αντικείμενα:

```
In [21]: a=10/3

In [22]: a.numerical_approx()
Out[22]: 3.33333333333333

In [23]: (10/3).numerical_approx()
Out[23]: 3.33333333333333
```

1.7 Επίλυση αλγεβρικών εξισώσεων

Η βασική συνάρτηση επίλυσης αλγεβρικών εξισώσεων είναι η ***solve***:

solve(eqnς, vars)	Επιλύει τις (την) εξισώσεις (-η) eqns ως προς τις (την) μεταβλητές (-η) vars.

Μία εξίσωση στο Sage εισάγεται με διπλό σύμβολο ισότητας ==, αφού το απλό σύμβολο της ισότητας χρησιμοποιείται από το Sage για αναθέσεις.

Οι ρίζες της εξίσωσης x_0, x_1, … εμφανίζονται σε λίστα υπό τη μορφή ισοτήτων "==", ως εξής:

$$[x{==}x_0,\ x{==}\ x_1,\ \ldots].$$

Όλες οι ρίζες της εξίσωσης εμφανίζονται μόνο μία φορά ανεξάρτητα από τη πολλαπλότητά τους. Αν μας ενδιαφέρει και η πολλαπλότητα της ρίζας τότε πρέπει να χρησιμοποιήσουμε τη μέθοδο ***roots( )***.

Αν έχουμε μία εξίσωση που το δεξιό μέλος της είναι 0, μπορούμε να το παραλείψουμε:

```
In [25]: solve(7*x + 3, x)
Out[25]: [x == (-3/7)]
```

Οι συμβολικές μεταβλητές πρέπει να οριστούν κατάλληλα στο SAGE (με εξαίρεση το x το οποίο έχει προκαθοριστεί ως συμβολική μεταβλητή). Ο τρόπος είναι ο εξής:

y = var (‘y’)	για μία συμβολική μεταβλητή
y, z = var(‘y z’)	για περισσότερες από μία συμβολική σειρά

Οπότε, αν έχουμε μία εξίσωση η οποία περιέχει δύο ή περισσότερες συμβολικές μεταβλητές, αρχικά πρέπει να ορίσουμε τις συμβολικές μεταβλητές και στη πρέπει να ορίσουμε υποχρεωτικά ως προς ποια μεταβλητή θέλουμε να τη λύσουμε:

```
In [5]: y, a, b, c, d = var ('y a b c d')

In [6]: solve(a*y + b == c*x + d, x)
Out[6]: [x == (a*y + b - d)/c]
```

Μπορούμε να λύσουμε την εξίσωση ως προς οποιαδήποτε μεταβλητή της:

```
In [7]: solve(a*y + b == c*x + d, y)
Out[7]: [y == (c*x - b + d)/a]

In [8]: solve(a*y + b == c*x + d, a)
Out[8]: [a == (c*x - b + d)/y]

In [9]: solve(a*y + b == c*x + d, b)
Out[9]: [b == c*x - a*y + d]
```

Στην περίπτωση που έχουμε συστήματα εξισώσεων, ο προσδιορισμός eqns είναι μία λίστα της μορφής [eqn1, eqn2, …] ενώ ο προσδιορισμός vars είναι μία λίστα της μορφής[var1, var2, ….]

```
In [10]: y, z = var('y z')

In [11]: solve([3*x+z==3, 7*x-9*y-2*z==1, x+y+7*z==3], [x,y,z])
Out[11]: [[x == (155/167), y == (94/167), z == (36/167)]]
```

Με τη συνάρτηση ***solve*** έχουμε ακριβής επίλυση μίας εξίσωσης:

```
In [12]: solve(x**2+5*x-3==0,x)
Out[12]: [x == -1/2*sqrt(37) - 5/2, x == 1/2*sqrt(37) - 5/2]
```

Αν θέλουμε τα αποτελέσματα μας να εμφανίζονται σε μία πιο μαθηματική μορφή θα πρέπει να χρησιμοποιήσουμε την εντολή %display typeset ή την pretty_print_default(True).

%display typeset ή pretty_print_default(True)	εμφανίζονται τα αποτελέσματα σε ποιο μαθηματική μορφή
pretty_print_default(False)	εμφανίζονται τα αποτελέσματα σε ποιο απλή μορφή

```
In [13]: %display typeset

In [14]: solve(x**2+5*x-3==0,x)
Out[14]:
```

$$\left[x = -\frac{1}{2}\sqrt{37} - \frac{5}{2}, x = \frac{1}{2}\sqrt{37} - \frac{5}{2}\right]$$

```
In [15]: pretty_print_default(False)

In [16]: solve(x**2+5*x-3==0,x)
Out[16]: [x == -1/2*sqrt(37) - 5/2, x == 1/2*sqrt(37) - 5/2]
```

Όταν έχουμε πολυωνυμικές εξισώσεις και μας ενδιαφέρει η αριθμητική (προσεγγιστική) επίλυση της εξίσωσης μπορούμε να χρησιμοποιήσουμε, τη μέθοδο ***roots( )***. Η μέθοδος ***roots( )*** επιστρέφει εκτός από τις ρίζες και την πολλαπλότητα της κάθε μιας. Επίσης στη μέθοδος ***roots( )*** μπορούμε να καθορίσουμε και τον δακτύλιο των λύσεων.

(eqn).roots(ring = XX)	λύνει αριθμητικά την εξίσωση eqn=0 στο δακτύλιο XX (ZZ ο δακτύλιος των ακεραίων, QQ ο δακτύλιος των ρητών, RR ο δακτύλιος των πραγματικών και CC ο δακτύλιος των μιγαδικών), επιστρέφοντας τις ρίζες μαζί με τη πολλαπλότητά τους.

Προσοχή στη μέθοδο ***roots( )*** χρησιμοποιούμε μόνο το δεξί μέλος μιας εξίσωσης της οποίας το αριστερό ισούται με μηδέν.

```
In [18]: (x**2+5*x-3).roots(ring=RR)

Out[18]: [(-5.54138126514911, 1), (0.541381265149110, 1)]
```

Με τη μέθοδο roots( ) μπορούμε να λύσουμε την ίδια εξίσωση σε διάφορους δακτυλίους λύσεων. Για παράδειγμα η λύση της εξίσωσης $x^6 - 2x^2 - 3x + 4 = 0$ στους ακεραίους αριθμούς δίνει μία λύση πολλαπλότητας 1:

```
In [21]: (x**6-2*x**2-3*x+4).roots(ring=ZZ)

Out[21]: [(1, 1)]
```

στους πραγματικούς αριθμούς δίνει 2 λύσεις πολλαπλότητας 1:

```
In [22]: (x**6-2*x**2-3*x+4).roots(ring=RR)

Out[22]: [(1.00000000000000, 1), (1.06916982139911, 1)]
```

και στους μιγαδικούς αριθμούς δίνει 6 λύσεις, όλες πολλαπλότητας 1:

```
In [23]: (x**6-2*x**2-3*x+4).roots(ring=CC)

Out[23]: [(1.00000000000000, 1),
          (1.06916982139911, 1),
          (-1.17674841649108 - 0.630259924714768*I, 1),
          (-1.17674841649108 + 0.630259924714768*I, 1),
          (0.142163505791523 - 1.44197149031700*I, 1),
          (0.142163505791523 + 1.44197149031700*I, 1)]
```

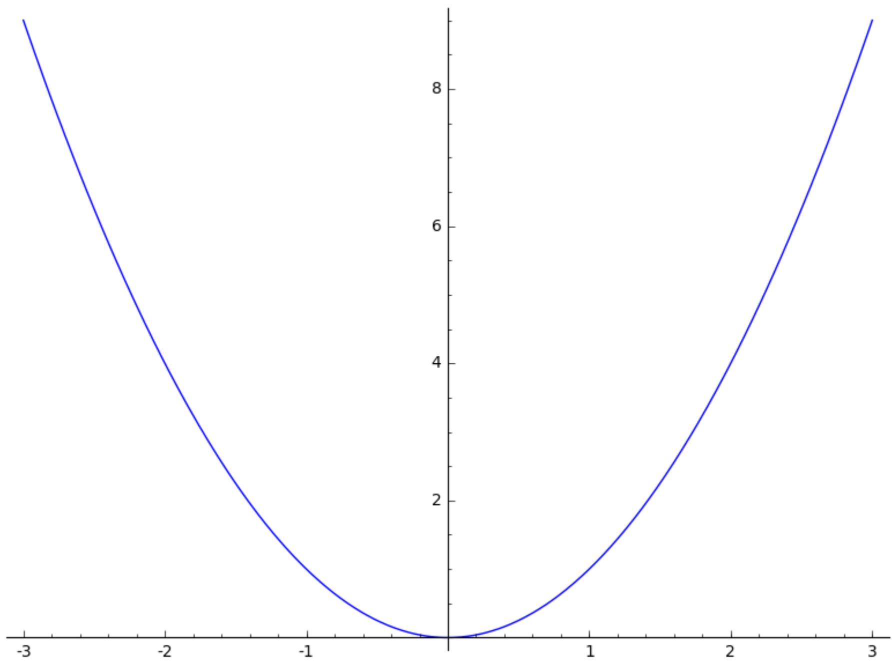
2. Δισδιάστατα γραφικά

2.1 Δισδιάστατες γραφικές παραστάσεις συναρτήσεων μίας μεταβλητής.

Η βασική εντολή σχεδίασης, του Sage, μιας γραφικής παράστασης μίας συνάρτησης μίας μεταβλητής είναι η συνάρτηση ***plot***.

plot(f(x), (x, xmin, xmax))	Σχεδιάζει τη γραφική παράσταση της συνάρτησης f(x) στο διάστημα (xmin, xmax).

In[1]: plot(x2, (x, -3, 3))**
Out[1]:

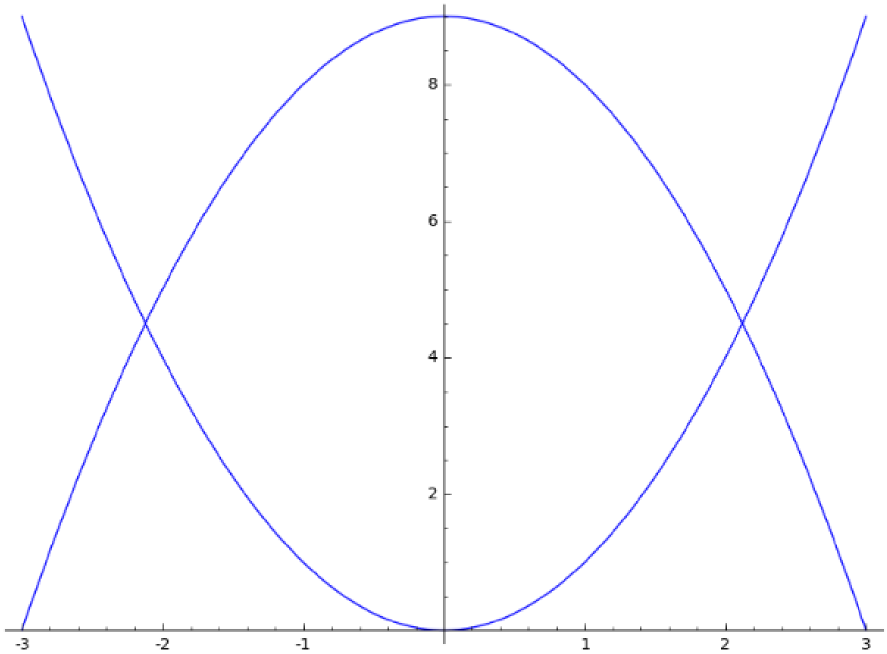


Έχουμε τη δυνατότητα να σχεδιάσουμε τις γραφικές παραστάσεις δύο ή περισσοτέρων συναρτήσεων με κοινό πεδίο ορισμού στο ίδιο σύστημα αξόνων.

plot([f(x), g(x), …], (x, xmin, xmax)]	Σχεδιάζει τις γραφικές παραστάσεις των συναρτήσεων f(x), g(x), …, στο διάστημα (xmin, xmax) στο ίδιο σύστημα αξόνων.

In[2]: plot([x2, 9-x^2], (x, -3, 3))**

Out[2]:=



Παρατηρούμε ότι το Sage, σχεδίασε τη γραφική παράσταση της κάθε συνάρτησης με ίδιο χρώμα. Θα δούμε στη συνέχεια πως μπορούμε εμείς να ορίσουμε διαφορετικό χρώμα για κάθε μία γραφική παράσταση.

Όπως είδαμε η συνάρτηση ***plot*** μπορεί να σχεδιάσει τις γραφικές παραστάσεις δύο ή περισσοτέρων συναρτήσεων με κοινό πεδίο ορισμού στο ίδιο σύστημα αξόνων. Τι γίνεται όμως στην περίπτωση οι συναρτήσεις των οποίων τις γραφικές παραστάσεις θέλουμε να σχεδιάσουμε ορίζονται σε διαφορετικά πεδία ορισμού. Για αυτή τη περίπτωση το Mathematica διαθέτει τη συνάρτηση ***show***.

show(g1 + g2 + …) **ή g1 + g2 + …**	Σχεδιάζει μαζί δύο ή περισσότερες γραφικές παραστάσεις συναρτήσεων με διαφορετικό πεδίο ορισμού στο ίδιο σύστημα αξόνων.

Έστω ότι θέλουμε να σχεδιάσουμε τη γραφική παράσταση της συνάρτησης $y = x^2 - 9$ στο διάστημα [-4, 4] και τη γραφική παράσταση της συνάρτησης $y = \sin x$ στο διάστημα [0, 2π], στο ίδιο σύστημα αξόνων. Αρχικά δημιουργούμε δύο αντικείμενα g1 και g2, ένα για κάθε γραφική παράσταση.

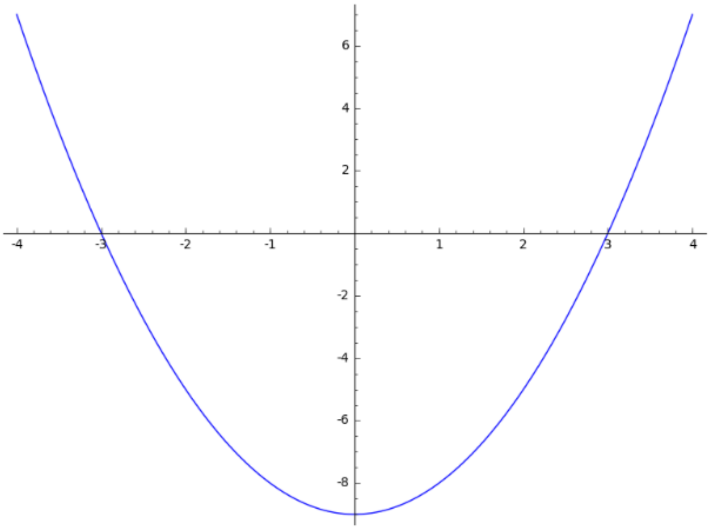
In[3]: g1 = plot((x^2 - 9), (x,- 4, 4))

In[4]: g2 = plot(sin(x), (x, 0, 2*pi))

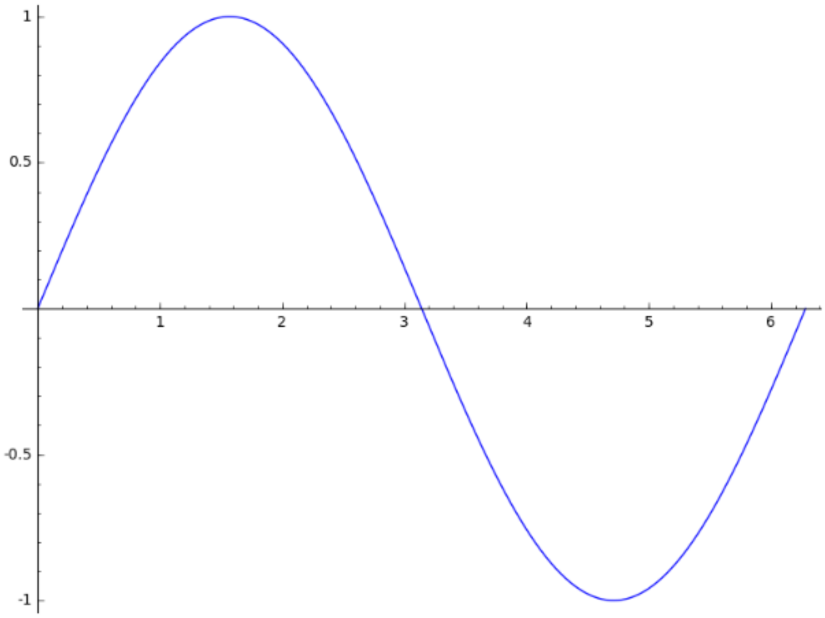
Παρατηρούμε ότι το Sage δεν σχεδιάζει τις γραφικές παραστάσεις. Αν θέλουμε να δούμε τις γραφικές παραστάσεις των παραπάνω συναρτήσεων θα πρέπει να χρησιμοποιήσουμε είτε τη συνάρτηση ***show*** είτε τη συνάρτηση ***plot***. Επίσης μπορούμε να χρησιμοποιήσουμε είτε τη μέθοδο ***show( )*** είτε τη μέθοδο ***plot( ).***

In[5]: plot(g1)

Out[5]:

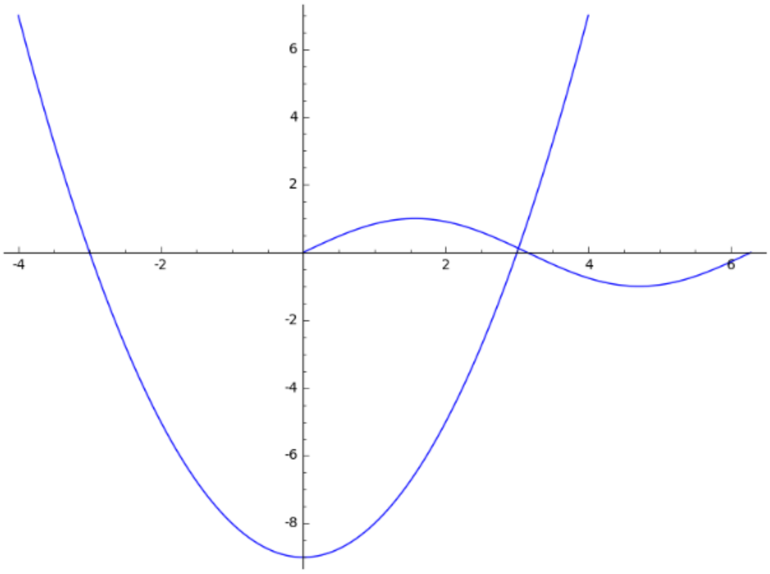


In[6]: g2.show()



Στη συνέχεια χρησιμοποιώντας την εντολή show σχεδιάζουμε, στο ίδιο σύστημα αξόνων, τις γραφικές παραστάσεις g1 και g2.

In[7]: show(g1 + g2)



Το ίδιο αποτέλεσμα θα είχαμε αν χρησιμοποιήσουμε τη συνάρτηση ***show*** με ορίσματα τη συνάρτηση ***plot***, δηλαδή χωρίς να απαιτείται να δημιουργήσουμε τα αντικείμενα g1 και g2.

In[8]: show(plot((x^2 - 9), (x,- 4, 4))+ plot(sin(x), (x, 0, 2*pi))

Επίσης παρατηρούμε ότι στη κλήση της συνάρτησης ***show*** στο αποτέλεσμα δεν εμφανίζεται η ετικέτα ***Out[ ]***.

2.2 Οι επιλογές των δισδιάστατων γραφικών

Το Sage, προκειμένου να αποδώσει καλύτερα τη γραφική παράσταση με αισθητικά ικανοποιητικό τρόπο, ρυθμίζει από μόνο του όλες τις λεπτομέρειες που θα οδηγήσουν στο αποτέλεσμα αυτό (π.χ. κλίμακα, τοποθέτηση αξόνων, εύρος πεδίου ορισμού, χρωματισμούς, κ.τ.λ.) και τις περισσότερες φορές το επιτυγχάνει.

Παρ' όλα αυτά το Sage μας δίνει την δυνατότητα να επηρεάσουμε το αποτέλεσμα με διάφορους τρόπους, ώστε να πάρουμε τη γραφική παράσταση που επιθυμούμε. Αυτό γίνεται παραθέτοντας στη σύνταξη της συνάρτησης ***plot***, οποιεσδήποτε από τις επιλογές που τη συνοδεύουν, και μάλιστα γράφοντας τις με οποιαδήποτε σειρά.

Η σύνταξη μίας επιλογής γίνεται ως εξής:

όνομα_επιλογής = τιμή.

Συνεπώς, αν χρησιμοποιήσουμε την εντολή ***plot*** με επιλογές, τότε η σύνταξή της γίνεται:

plot(f(x), (x, xmin, xmax), επιλογή1 = τιμή1, επιλογή2 = τιμή 2, …).

Κάθε επιλογή έχει μία προκαθορισμένη τιμή (προεπιλογή), την οποία παίρνει αυτόματα από το Sage εφόσον δεν ζητήσουμε κάτι διαφορετικό. Τις επιλογές, μαζί με τις προκαθορισμένες τιμές (προεπιλογές) του, της συνάρτησης ***plot***, μπορούμε να τις δούμε αν πληκτρολογήσουμε την εντολή ***plot.options***:

In[9]: plot.options

Out[9]:

```
{'adaptive_recursion': 5,
 'adaptive_tolerance': 0.01,
 'alpha': 1,
 'aspect_ratio': 'automatic',
 'detect_poles': False,
 'exclude': None,
 'fill': False,
 'fillalpha': 0.5,
 'fillcolor': 'automatic',
 'legend_label': None,
 'plot_points': 200,
 'rgbcolor': (0, 0, 1),
 'thickness': 1}
```

Στη συνέχεια θα εξετάσουμε μερικές από τις επιλογές που χρησιμοποιούνται συχνότερα.

2.2.1 ymin, ymax

Οι επιλογές ***ymin*** και ***ymax*** καθορίζουν το εύρος του άξονα y, οπότε καθορίσουν τα σημεία που πρέπει να συμπεριληφθούν στη γραφική παράσταση.

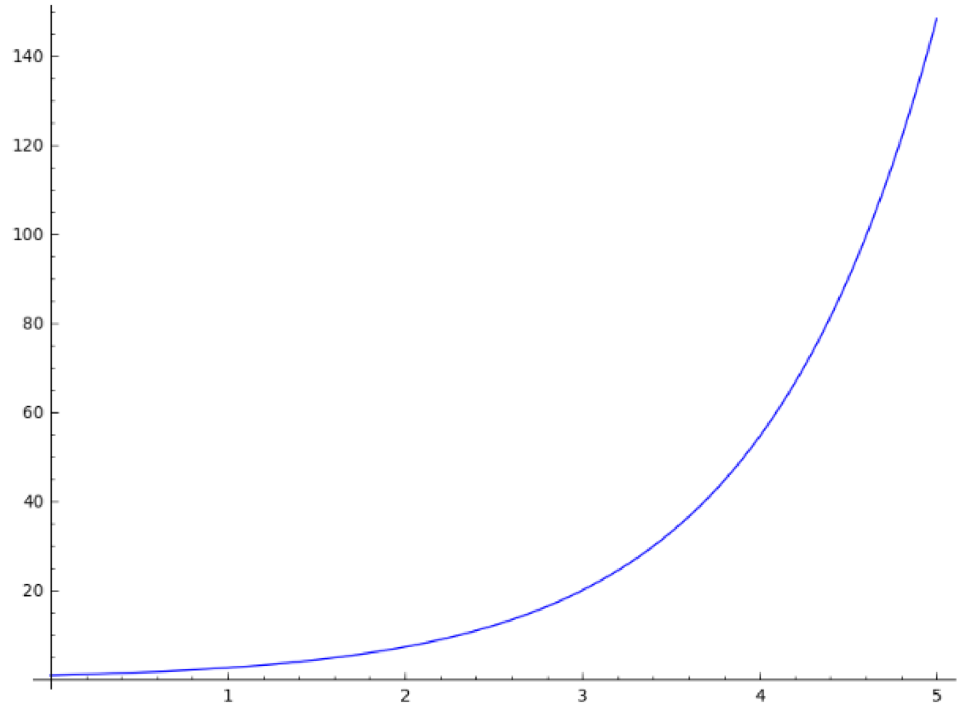
ymin=y1	καθορίζει ότι το κάτω άκρο του άξονα yy’ είναι το y1.
ymax=y2	καθορίζει ότι το άνω άκρο του άξονα yy’ είναι το y2.
ymin =y1, ymax=y2	απεικονίζει μόνο τα σημεία που οι τιμές οι τιμές των τεταγμένων τους βρίσκονται μεταξύ y1 και y2.

Εξ’ ορισμού, οι προεπιλεγμένες τιμές, των ***ymin*** και ***ymax*** καθορίζονται από τις τιμές της συνάρτησης για όλο το πεδίο ορισμού της.

Για παράδειγμα, έστω ότι, θέλουμε να σχεδιάσουμε τη συνάρτηση $f(x) = e^x$ με πεδίο ορισμού το [0, 5]. Αν δεν χρησιμοποιήσουμε τις επιλογές ***ymin*** και ***ymax***, τότε το εύρος του κάθετου άξονα yy’ καθορίζεται από τις τιμές της συνάρτησης για το πεδίο ορισμού της, δηλαδή από $e^0 = 1$ έως $e^5 \approx 148.413159$.

In[10]: plot(exp(x), (x, 0, 5))

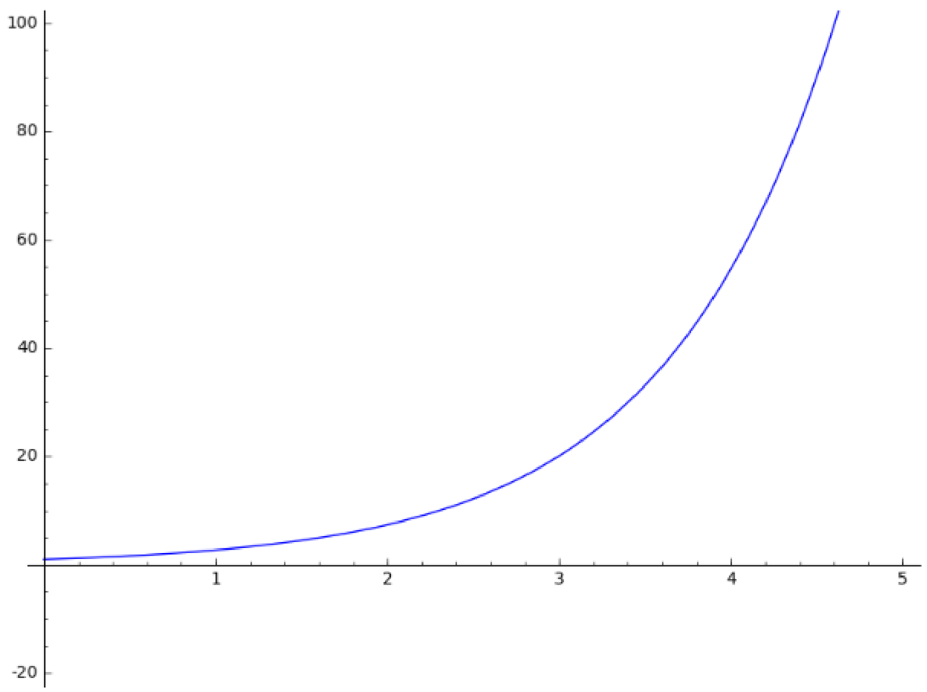
Out[10]:



Μπορούμε όμως χρησιμοποιώντας τις επιλογές ***ymin*** και ***ymax***, να αλλάξουμε το εύρος του κάθετου άξονα γγ'.

In[11]: plot(exp(x), (x, 0, 5) , ymin=-20, ymax=100)

Out[11]:



Παρατηρούμε ότι ενώ το πεδίο ορισμού της συνάρτησης δεν έχει αλλάξει, εμφανίζονται μόνο τα σημεία που οι τεταγμένες τους κυμαίνονται από -20 έως 100.

2.2.2 aspect_ratio

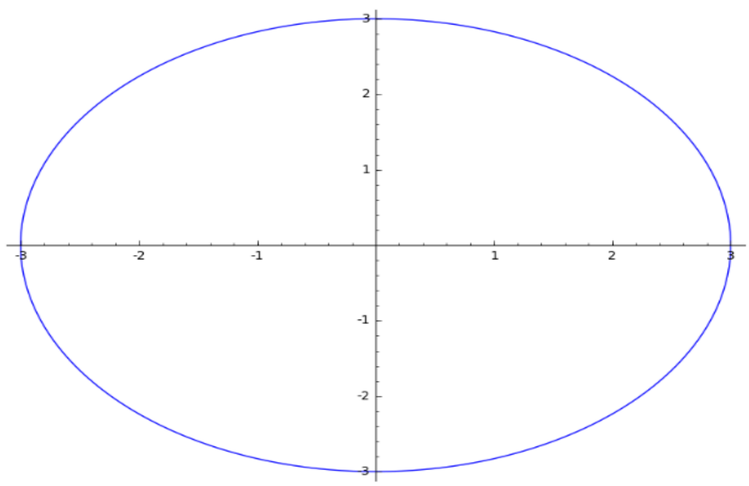
Κατά τη σχεδίαση γραφικών παραστάσεων, παρατηρούμε ότι το μήκος του οριζόντιου άξονα και του κατακόρυφου άξονα δεν είναι το ίδιο. Η επιλογή ***aspect_ratio*** καθορίζει την αναλογία του μήκους του κατακόρυφου άξονα (ύψος γραφικής παράστασης) προς το μήκος του οριζόντιου άξονα (πλάτος γραφικής παράστασης) και παίρνει τις εξής τιμές:

aspect_ratio = ‘automatic’ (προεπιλογή)	η αναλογία ύψους προς πλάτος της γραφικής παράσταση είναι ίση με $\frac{2}{1+\sqrt{5}}=0.618934$ (αντίστροφη της χρυσής τομής)
aspect_ratio = 1	καθορίζει το λόγο του ύψους προς το πλάτος της γραφικής παράστασης λαμβάνοντας υπόψη τις πραγματικές τιμές των συντεταγμένων των σημείων της γραφικής παράστασης
aspect_ratio = n	ορίζει το λόγο του ύψους προς το πλάτος της γραφικής παράστασης ίσο με τιμή n.

Η παρακάτω εντολή σχεδιάζει έναν κύκλο με ακτίνα 3 και κέντρο την αρχή των αξόνων:

In[12]:= plot([-sqrt(9-x2),sqrt(9-x**2)],(x, -3, 3))**

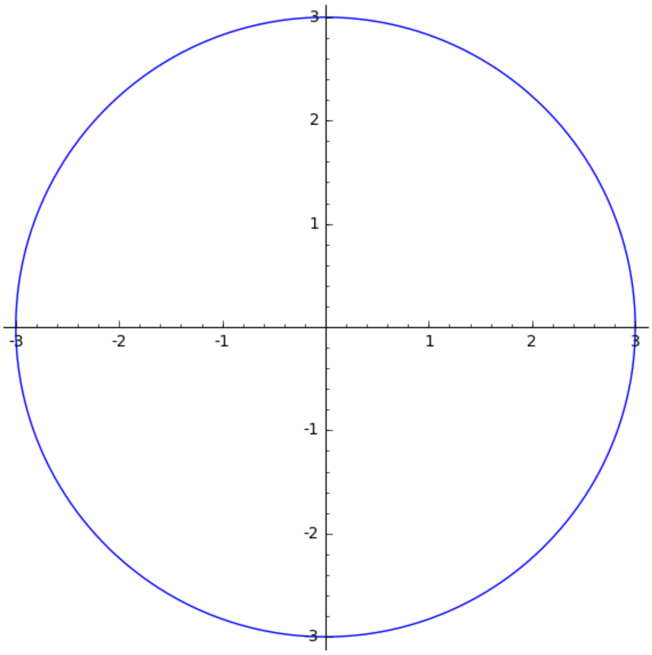
Out[12]:=



Παρατηρούμε ότι, επειδή ο λόγος του ύψους προς το πλάτος της γραφικής παράστασης είναι ο προκαθορισμένος, η γραφική παράσταση έχει τη μορφή έλλειψης. Για να εμφανιστεί ο κύκλος στρογγυλός, θα πρέπει να χρησιμοποιήσουμε την επιλογή **aspect_ratio**:

In[13]:= plot([-sqrt(9-x2),sqrt(9-x**2)],(x, -3, 3), aspect_ratio=1)**

Out[13]:=

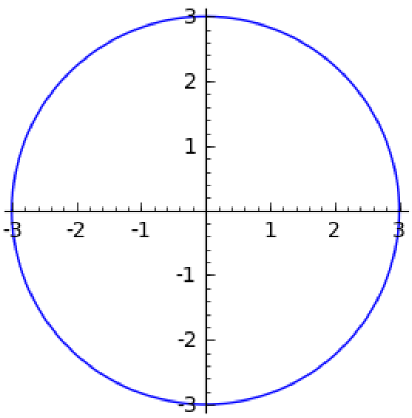


2.2.3 figsize

figsize = n	καθορίζει το μέγεθος του τετράγωνου πλαισίου σχεδίασης. Εξ' ορισμού είναι figsize = 8.

In[14]:= plot([-sqrt(9-x2),sqrt(9-x**2)],(x, -3, 3), aspect_ratio=1, figsize=4)**

Out[14]:=



Οι επόμενες επιλογές καθορίζουν το στυλ της γραμμής της γραφικής παράστασης

2.2.4 color

Η επιλογή ***color*** καθορίζει το χρώμα της γραμμής.

color = (r, g, b) ή rgbcolor = (r, g, b)	καθορίζει το χρώμα ως συνδυασμός τριών χρωμάτων, του κόκκινου (r), του πράσινου (g) και του μπλε (b). Δίνοντας σε κάθε ένα από τα r, g, b, τιμές στο διάστημα [0, 1]. Εξ' ορισμού έχει την τιμή (0, 0, 1) που αντιστοιχεί στο μπλε χρώμα.
color = string	το χρώμα καθορίζεται με τη χρήση μίας συμβολοσειράς π.χ. 'red', 'blue', 'purple' κ.τ.λ.

Με χρήση της συνάρτησης sorted(colors) εμφανίζονται τα οι συμβολοσειρές (ονόματα) για τα χρώματα που διαθέτει το Sage ταξινομημένα αλφαβητικά.

In[15]: sorted(colors)

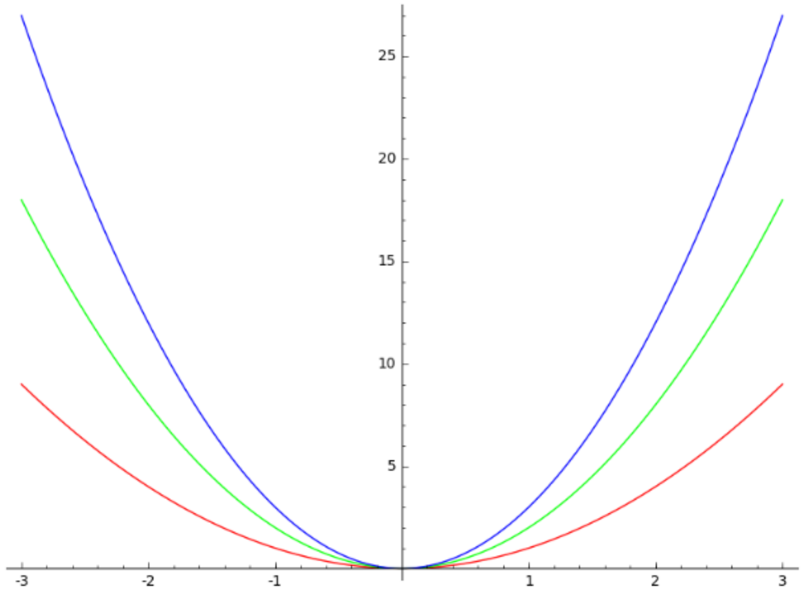
Out[15]:

```
['aliceblue',
 'antiquewhite',
 'aqua',
 'aquamarine',
 'automatic',
 'azure',
 'beige',
 'bisque',
 'black',
 'blanchedalmond',
 'blue',
 'blueviolet',
 'brown',
 'burlywood',
 'cadetblue',
 'chartreuse',
 'chocolate',
 'coral',
 'cornflowerblue',
```

In[16]:

```
g1=plot(x^2,(x,-3,3), color=(1,0,0))
g2=plot(2*x^2,(x,-3,3), color=(0,1,0))
g3=plot(3*x^2,(x,-3,3), color='blue')
g1+g2+g3
```

Out[16]:=



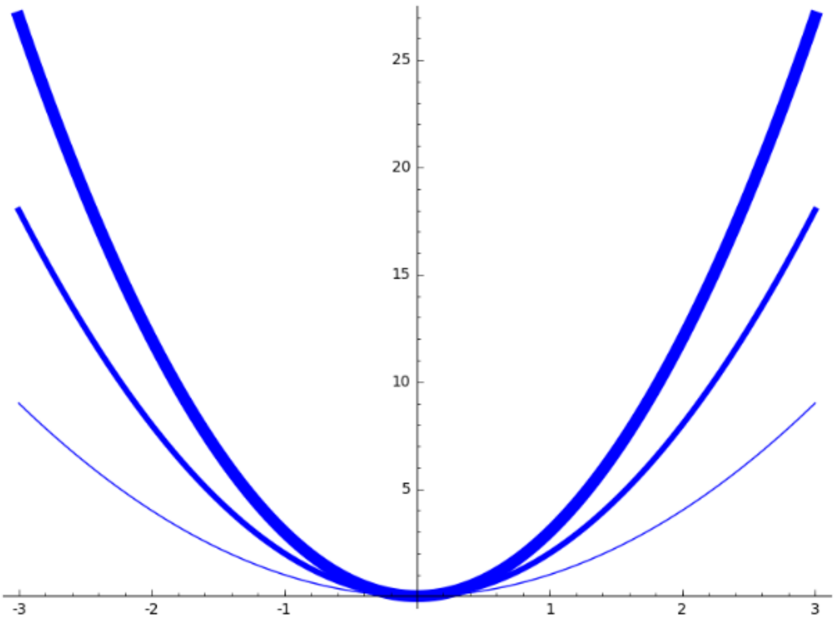
2.2.5 thickness

Η επιλογή ***thickness*** καθορίζει το πάχος της γραμμής.

thickness = n	καθορίζει ότι η γραμμή είναι n φορές παχύτερη. Εξ΄ ορισμού η τιμή της thickness είναι 1.

In[17]:
```
g1=plot(x^2,(x,-3,3))
g2=plot(2*x^2,(x,-3,3), thickness=4)
g3=plot(3*x^2,(x,-3,3), thickness=8)
g1+g2+g3
```

Out[17]:



2.2.6 linestyle

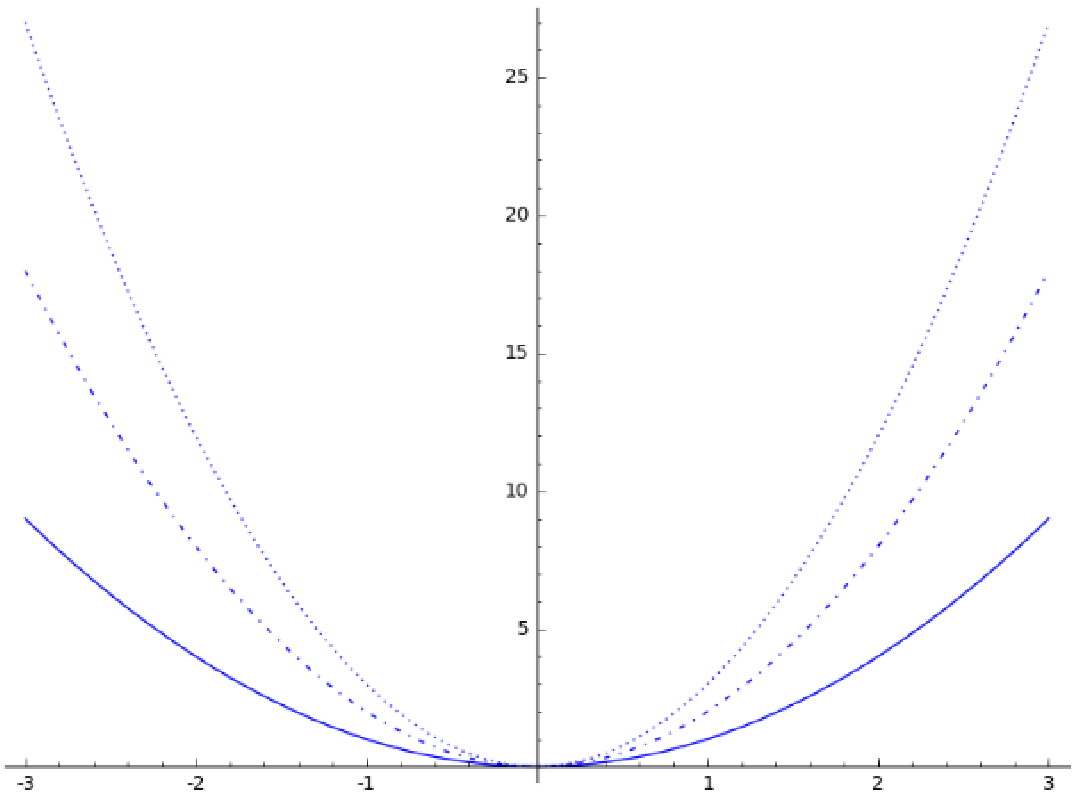
Η επιλογή ***linestyle*** καθορίζει το είδος της γραμμής. Το είδος της γραμμής μπορεί να είναι ένα από τα παρακάτω:

linestyle = "-" ή "solid"	σχεδιάζει συνεχόμενη γραμμή.
linestyle = "--" ή "dashed"	σχεδιάζει διακεκομμένη γραμμή.
linestyle = "-." ή "dashdot"	σχεδιάζει διακεκομμένη γραμμή με ενδιάμεσα τελείες.
linestyle = ":" ή "dotted"	σχεδιάζει διακεκομμένη γραμμή με τελείες.
linestyle = " " ή "None"	δεν σχεδιάζει γραμμή.

Εξ' ορισμού το είδος της γραμμής είναι συνεχόμενη γραμμή.

```
In[18]: g1=plot(x^2,(x,-3,3))
        g2=plot(2*x^2,(x,-3,3), linestyle="-.")
        g3=plot(3*x^2,(x,-3,3), linestyle="dotted")
        g1+g2+g3
```

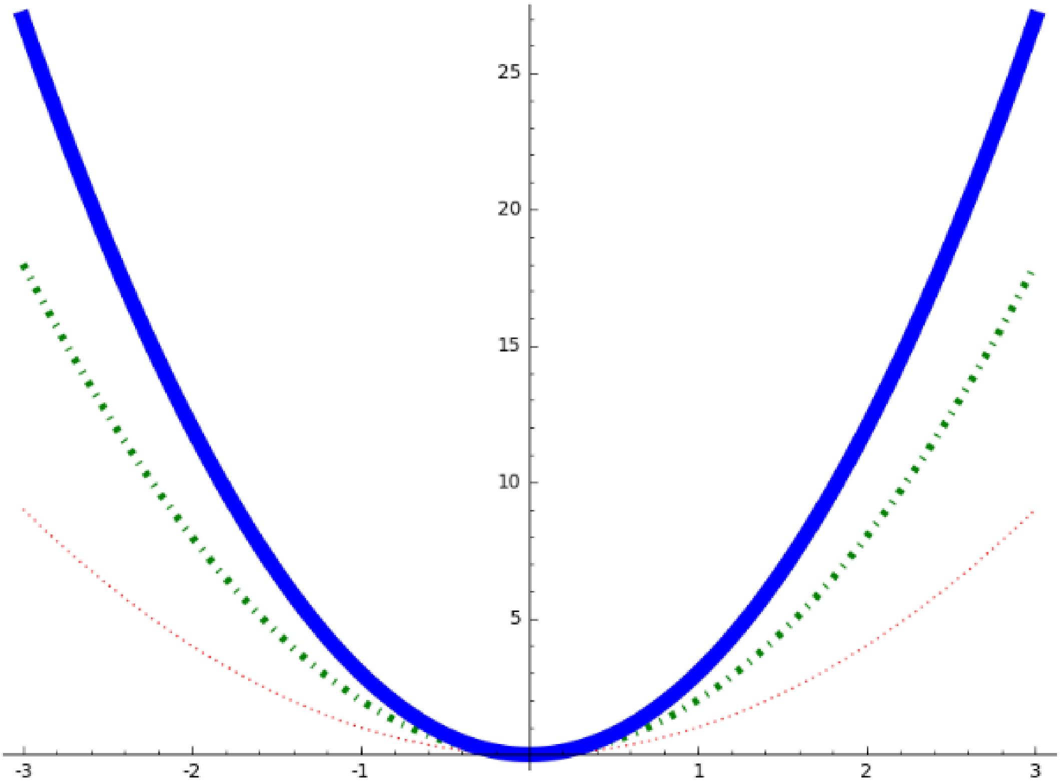
Out[18]:



Ας συνδυάσουμε όλες τις παραπάνω επιλογές που καθορίζουν το στυλ της γραμμής της γραφικής παράστασης:

In[19]: g1=plot(x^2,(x,-3,3), color=(1,0,0), linestyle="dotted")
g2=plot(2*x^2,(x,-3,3), color="green", thickness=2, linestyle="-.")
g3=plot(3*x^2,(x,-3,3), thickness=4)
g1+g2+g3

Out[19]:



2.2.7 Τίτλος στη γραφική παράσταση

Η επιλογή ***title*** τοποθετεί μία επικεφαλίδα (τίτλο) στη γραφική παράσταση.

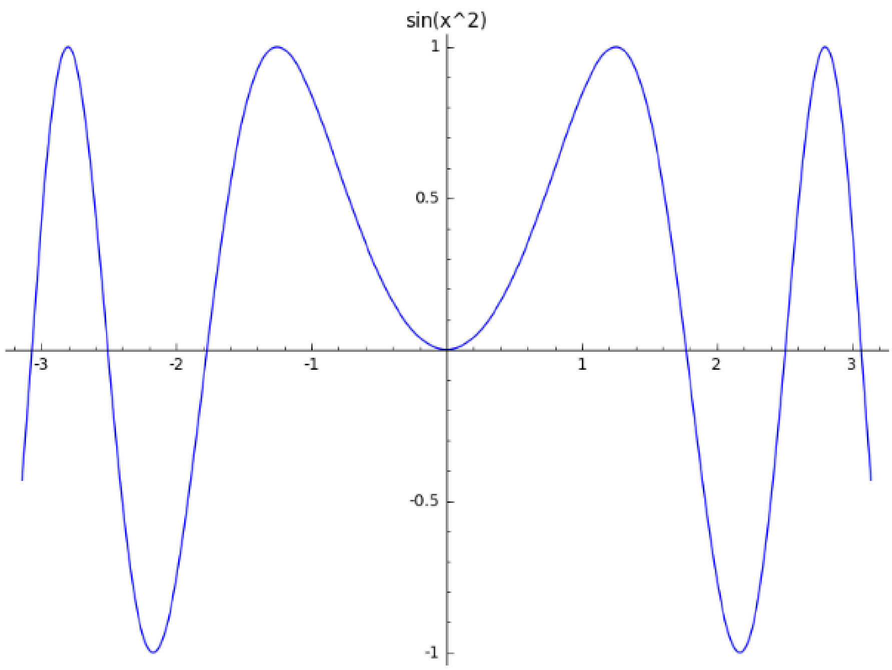
title = None (προεπιλογή)	δεν θέτει καμία επικεφαλίδα
title = 'επικεφαλίδα'	θέτει ως επικεφαλίδα στη γραφική παράσταση το κείμενο που ακολουθεί.

Η επιλογή ***title_pos*** τοποθετεί την επικεφαλίδα (τίτλο) σε συγκεκριμένη θέση στη γραφική παράσταση.

title_pos = None (προεπιλογή)	θέτει την επικεφαλίδα, πάνω και στο κέντρο της γραφικής παράστασης. Αντιστοιχεί στο ζεύγος (0.5, 1)
title_pos = (x_pos, y_pos)	θέτει την επικεφαλίδα στη γραφική παράσταση στη θέση (x_pos, y_pos). Η τιμή του x_pos βρίσκεται στο διάστημα [0, 1]. Το 0 αντιστοιχεί στο αριστερό άκρο του άξονα xx', το 0.5 στο κέντρο του άξονα xx' και το 1 στο δεξί άκρο του άξονα xx'. Η τιμή του y_pos βρίσκεται στο διάστημα [0, 1]. Το 0 αντιστοιχεί στο κάτω άκρο του άξονα yy', το 0.5 στο κέντρο του άξονα yy' και το 1 στο πάνω άκρο του άξονα yy'.

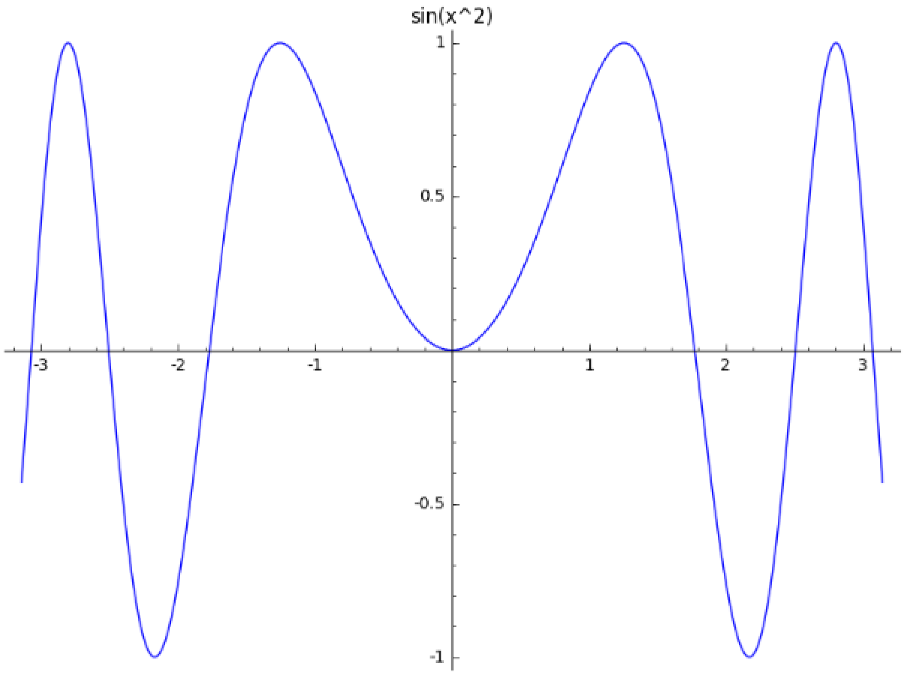
In[1]: plot(sin(x2), (x, -pi, pi), title='sin(x^2)')**

Out[1]:



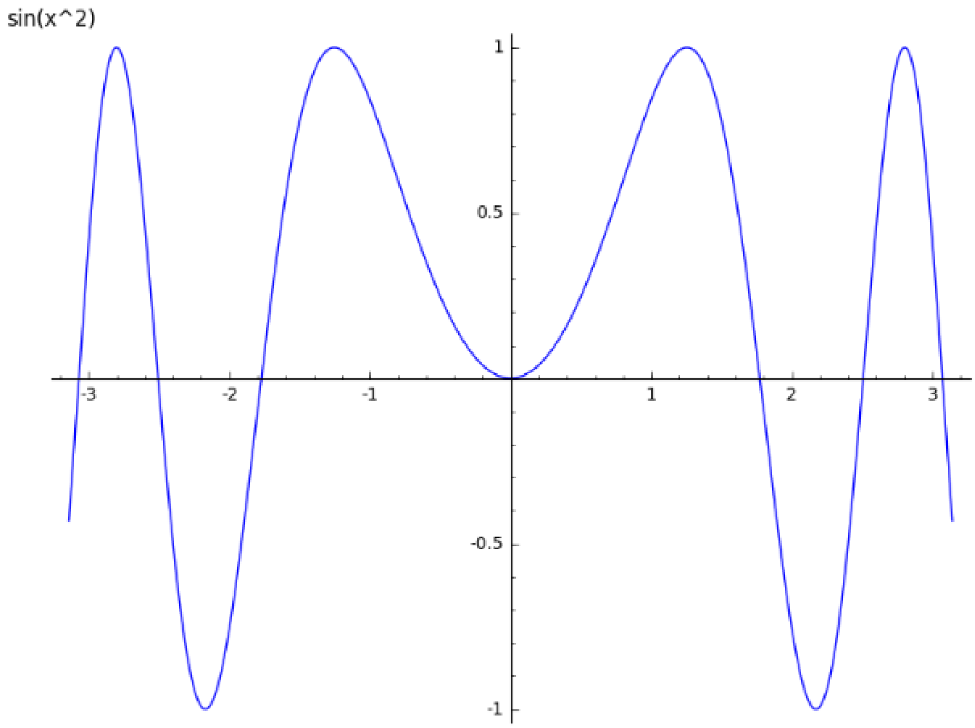
In[2]: plot(sin(x2), (x, -pi, pi), title='sin(x^2)', title_pos=(0.5, 1))**

Out[2]:



In[3]: plot(sin(x2), (x, -pi, pi), title='sin(x^2)', title_pos=(0, 1))**

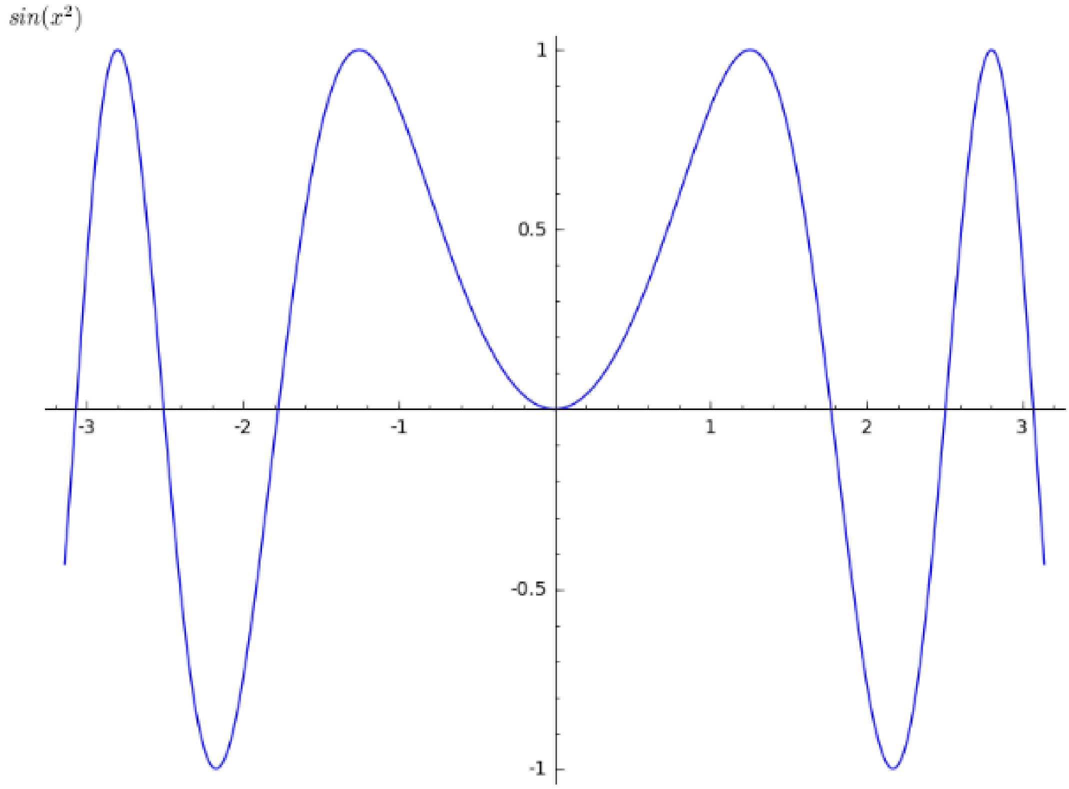
Out[3]:



Μπορούμε να χρησιμοποιήσουμε τη σύνταξη του Latex για πιο μαθηματική έκφραση του τίτλου:

In[4]: plot(sin(x2), (x, -pi, pi), title='$sin(x^2)$', title_pos=(0, 1))**

Out[4]:



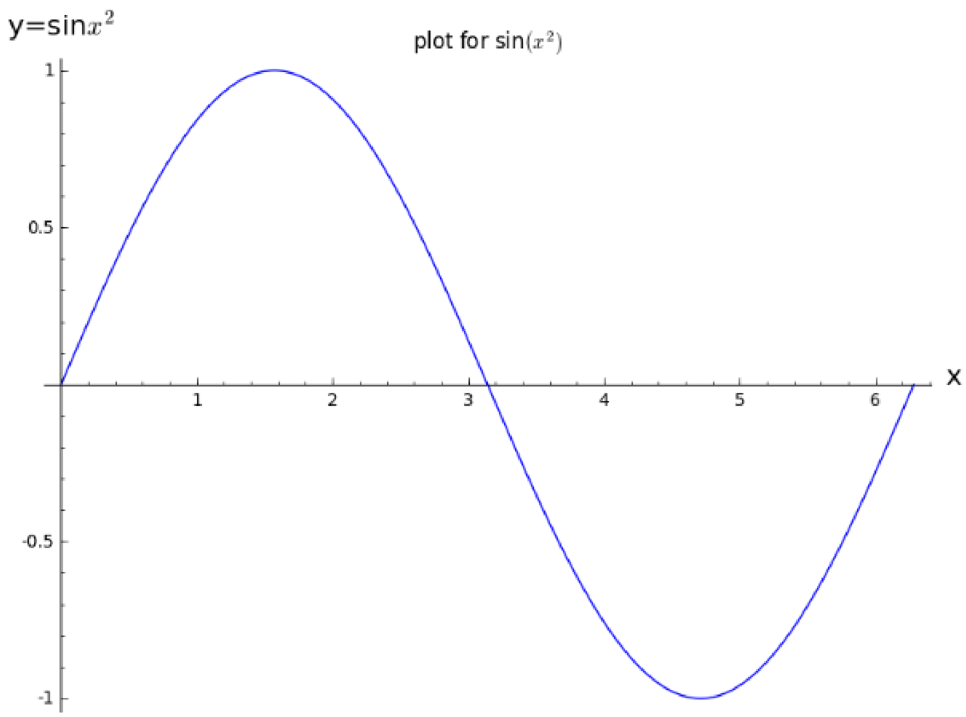
2.2.8 Τίτλος στους άξονες

Η επιλογή ***axes_labels*** καθορίζει, αν θα προστεθεί ετικέτα (τίτλος) σε κάποιον από τους άξονες, και στους δύο ή σε κανέναν.

axes_labels = None (προεπιλογή)	δεν θέτει ετικέτα σε κανέναν άξονα
axes_labels = [‘ετικέτα1’, ‘ετικέτα2’]	θέτει ως ετικέτα στον άξονα x το πρώτο κείμενο ενώ ως ετικέτα στον άξονα y το δεύτερο κείμενο. Ετικέτα μπορεί να είναι και το κενό.

In[5]:=plot(sin(x), (x, 0, 2*pi), title='plot for sin(x^2)', title_pos=(0.5, 1), axes_labels=['x', 'y=sinx^2'])

Out[5]:=



Η επιλογή ***axes_labels_size*** καθορίζει το μέγεθος των χαρακτήρων των ετικετών των αξόνων.

axes_labels_size = None (προεπιλογή)	για τους χαρακτήρες των ετικετών των αξόνων χρησιμοποιείται το προκαθορισμένο μέγεθος, που είναι 1.6 πολλαπλάσιο του προκαθορισμένου μεγέθους χαρακτήρων
axes_labels_size = s	όπου s ένας πραγματικός αριθμός, ο οποίος δίνει το σχετικό μέγεθος των χαρακτήρων ως προς το προκαθορισμένο μέγεθος χαρακτήρων.

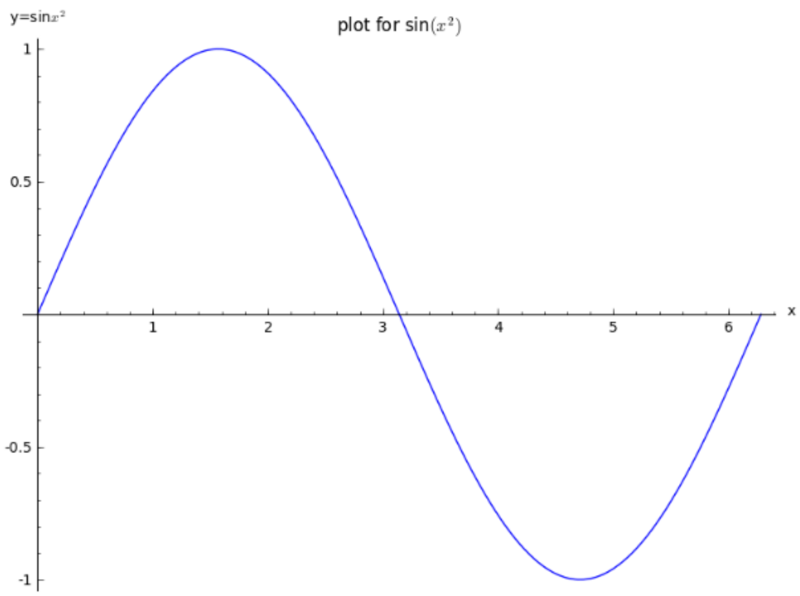
In[6]:= p1=plot(sin(x), (x, 0, 2*pi), title='plot for sin(x^2)', title_pos=(0.5, 1), axes_labels=['x', 'y=sinx^2'])

In[7]: p1.axes_labels_size()

Out[7]:= 1.6

In[8]:= plot(sin(x), (x, 0, 2*pi), title='plot for sin(x^2)', title_pos=(0.5, 1), axes_labels=['x', 'y=sinx^2'],axes_labels_size=1)

Out[8]:



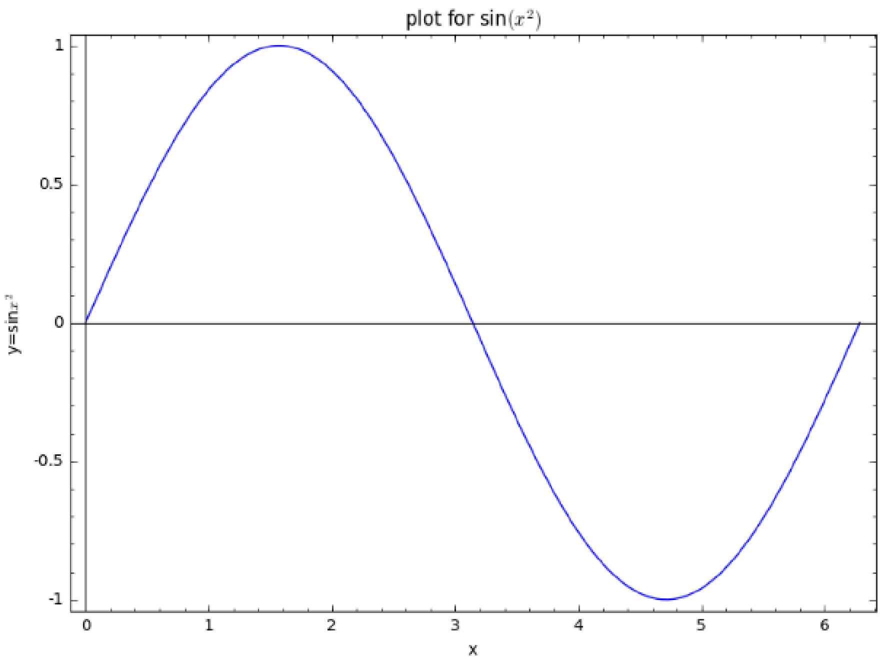
2.2.9 frame

Η επιλογή ***frame*** καθορίζει, αν η γραφική παράσταση θα περιβάλλεται από πλαίσιο.

frame = False (προεπιλογή)	ορίζει ότι η γραφική παράσταση θα σχεδιαστεί χωρίς πλαίσιο
frame = True	θέτει τη γραφική παράσταση σε πλαίσιο.

In[9]: plot(sin(x), (x, 0, 2*pi), title='plot for sin(x^2)', title_pos=(0.5, 1), axes_labels=['x', 'y=sinx^2'],axes_labels_size=1,frame=True)

Out[9]:



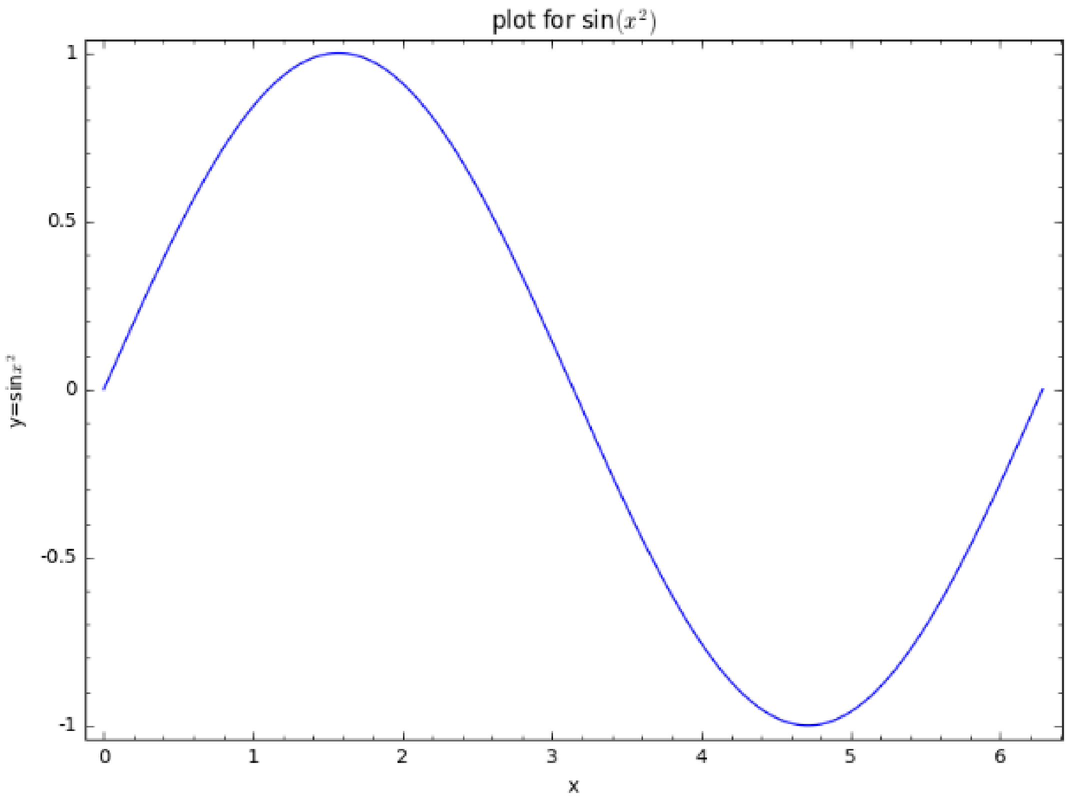
2.2.10 axes

Η επιλογή **axes** καθορίζει, αν θα σχεδιαστούν στη γραφική παράσταση οι δύο άξονες, ένας ή κανένας.

axes = True (προεπιλογή)	ορίζει ότι θα σχεδιαστούν και οι δύο άξονες
axes = False	ορίζει ότι δεν θα σχεδιαστεί κανένας από τους δύο άξονες.

In[10]: plot(sin(x), (x, 0, 2*pi), title='plot for sin(x^2)', title_pos=(0.5, 1), axes_labels=['x', 'y=sinx^2'],axes_labels_size=1,frame=True, axes=False)

Out[10]:



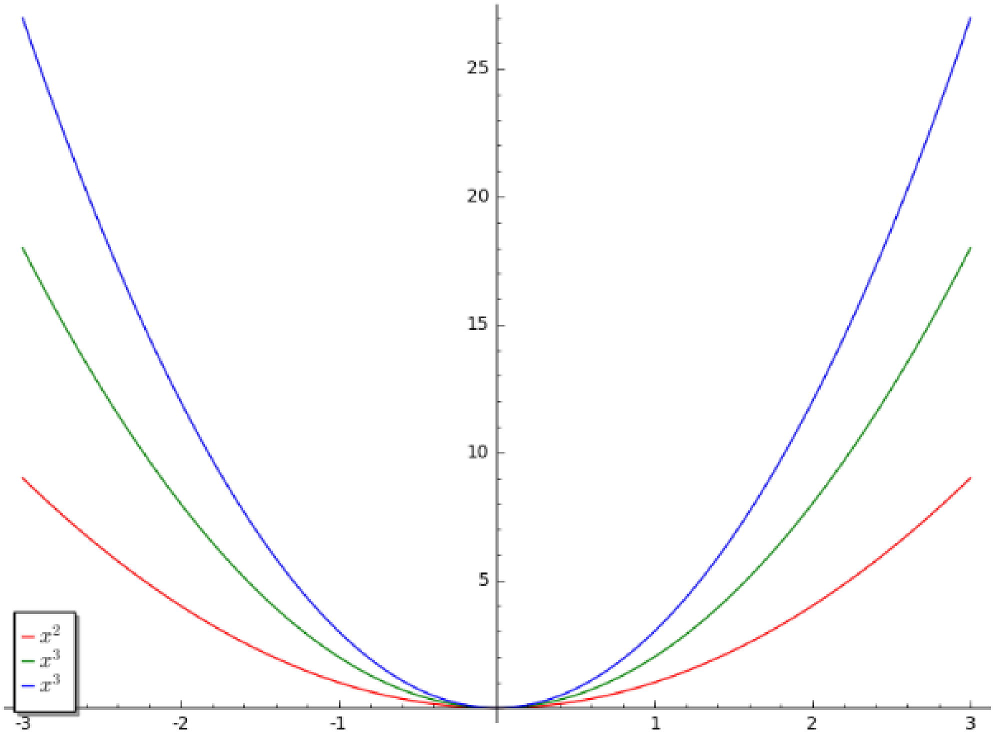
2.2.11 legend_label

Η επιλογή **legend_label** προσθέτει πλαίσιο με ετικέτα (λεζάντα) για τη γραφική παράσταση.

legend_label = None (προεπιλογή)	δεν προσθέτει λεζάντα για τη γραφική παράσταση
legend_label = 'κείμενο'	επισυνάπτει το κείμενο ως λεζάντα στη γραφική παράσταση.

```
In[11]:= g1=plot(x^2,(x,-3,3), color=(1,0,0), legend_label='$x^2$')
         g2=plot(2*x^2,(x,-3,3), color="green", legend_label='$2*x^2$')
         g3=plot(3*x^2,(x,-3,3), legend_label='$3*x^2$')
         g1+g2+g3
```

Out[11]:=



Η επιλογή ***legend_loc*** της συνάρτησης ***show*** καθορίζει τη θέση του πλαισίου με τις λεζάντες.

legend_loc = None (προεπιλογή)	το πλαίσιο με τις λεζάντες θέτεται κάτω αριστερά.
legend_loc = (x_pos, y_pos)	θέτει το πλαίσιο των λεζάντων στη θέση (x_pos, y_pos). Η τιμή του x_pos βρίσκεται στο διάστημα [0, 1]. Το 0 αντιστοιχεί στο αριστερό άκρο του άξονα xx’, το 0.5 στο κέντρο του άξονα xx’ και το 1 στο δεξί άκρο του άξονα xx’. Η τιμή του y_pos βρίσκεται στο διάστημα [0, 1]. Το 0 αντιστοιχεί στο κάτω άκρο του άξονα yy’, το 0.5 στο κέντρο του άξονα yy’ και το 1 στο πάνω άκρο του άξονα yy’.

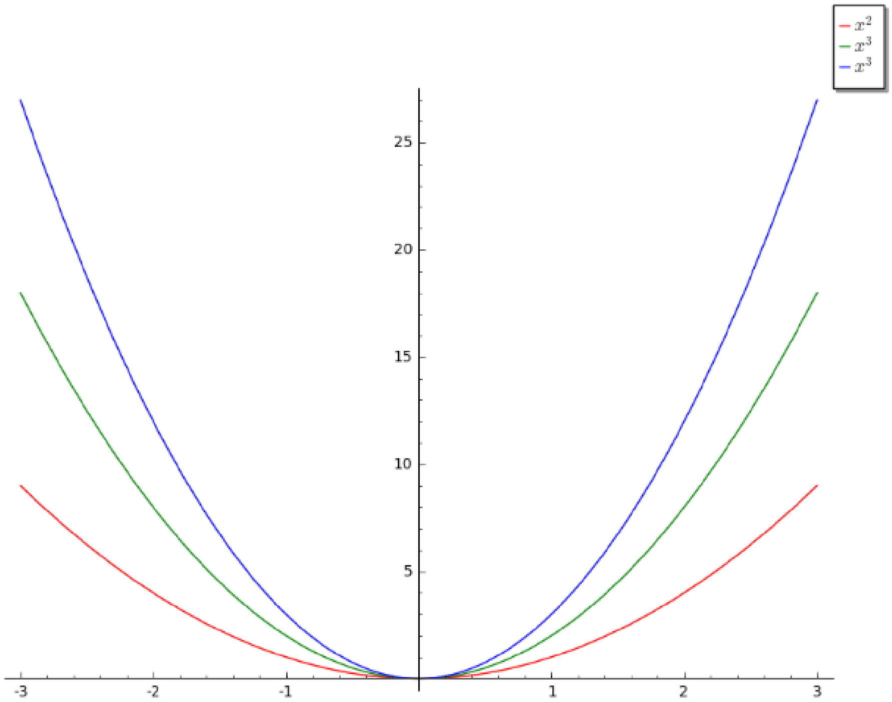
In[12]:= g1=plot(x^2,(x,-3,3), color=(1,0,0), legend_label='x^2')

g2=plot(2*x^2,(x,-3,3), color="green", legend_label='$2*x^2$')

g3=plot(3*x^2,(x,-3,3), legend_label='$3*x^2$')

show(g1+g2+g3,legend_loc=(1,1))

Out[12]:=



Με την επιλογή ***legend_color*** καθορίζουμε το χρώμα των ετικετών στο πλαίσιο.

legend_color = None (προεπιλογή)	το προκαθορισμένο χρώμα είναι το μαύρο (0, 0 ,0)
legend_color = (r, g, b)	καθορίζει το χρώμα ως συνδυασμός τριών χρωμάτων, του κόκκινου (r), του πράσινου (g) και του μπλε (b). Δίνοντας σε κάθε ένα από τα r, g, b, τιμές στο διάστημα [0, 1]
legend_color = string	το χρώμα καθορίζεται με τη χρήση μίας συμβολοσειράς π.χ. ‘red’, ‘blue’, ‘purple’ κ.τ.λ.

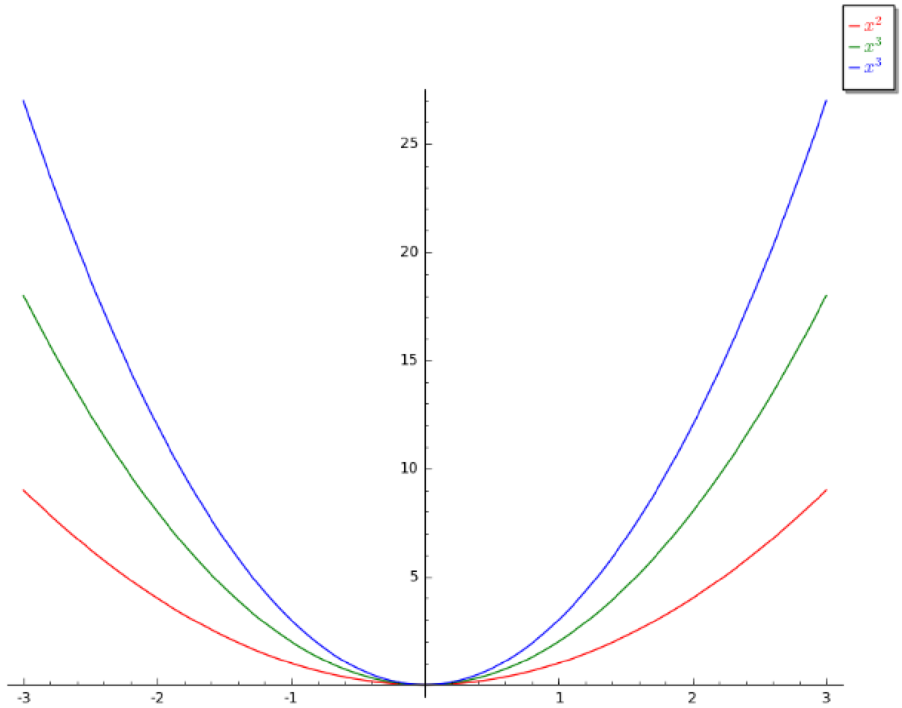
In[13]:= g1=plot(x^2,(x,-3,3), color=(1,0,0), legend_label='x^2', legend_color=(1,0,0))

g2=plot(2*x^2,(x,-3,3), color='green', legend_label='$2*x^2$', legend_color='green')

g3=plot(3*x^2,(x,-3,3), legend_label='$3*x^2$', legend_color=(0,0,1))

show(g1+g2+g3,legend_loc=(1,1))

Out[13]:=



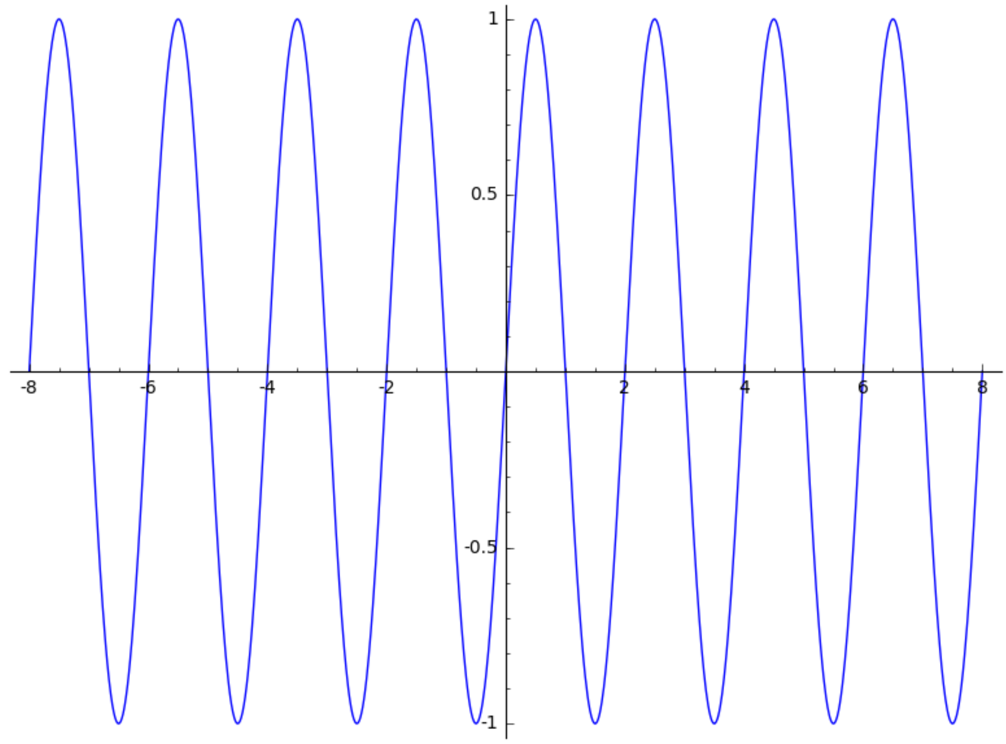
2.2.12 ticks

Η επιλογή ***ticks*** καθορίζει τις τιμές που εμφανίζονται στους άξονες xx' και yy'.

ticks = None (προεπιλογή)	οι τιμές καθορίζονται αυτόματα από το sage
ticks = s	οι τιμές στον άξονα xx' εμφανίζονται με βήμα s
ticks = [s ,t]	οι τιμές στον άξονα xx' εμφανίζονται με βήμα s, ενώ οι τιμές στον άξονα yy' εμφανίζονται με βήμα t
ticks = [[s1, s2, …], None]	μόνο οι τιμές s1, s2, … εμφανίζονται στον άξονα xx', ενώ στο άξονα yy' οι τιμές καθορίζονται αυτόματα
ticks = [None, [t1, t2, …]]	μόνο οι τιμές t1, t2, … εμφανίζονται στον άξονα yy', ενώ στο άξονα xx' οι τιμές καθορίζονται αυτόματα
ticks = [[…], […]]	τα στοιχεία της πρώτης λίστας εμφανίζονται ως τιμές στον άξονα xx' ενώ τα στοιχεία της δεύτερης λίστας εμφανίζονται ως τιμές στον άξονα yy'.

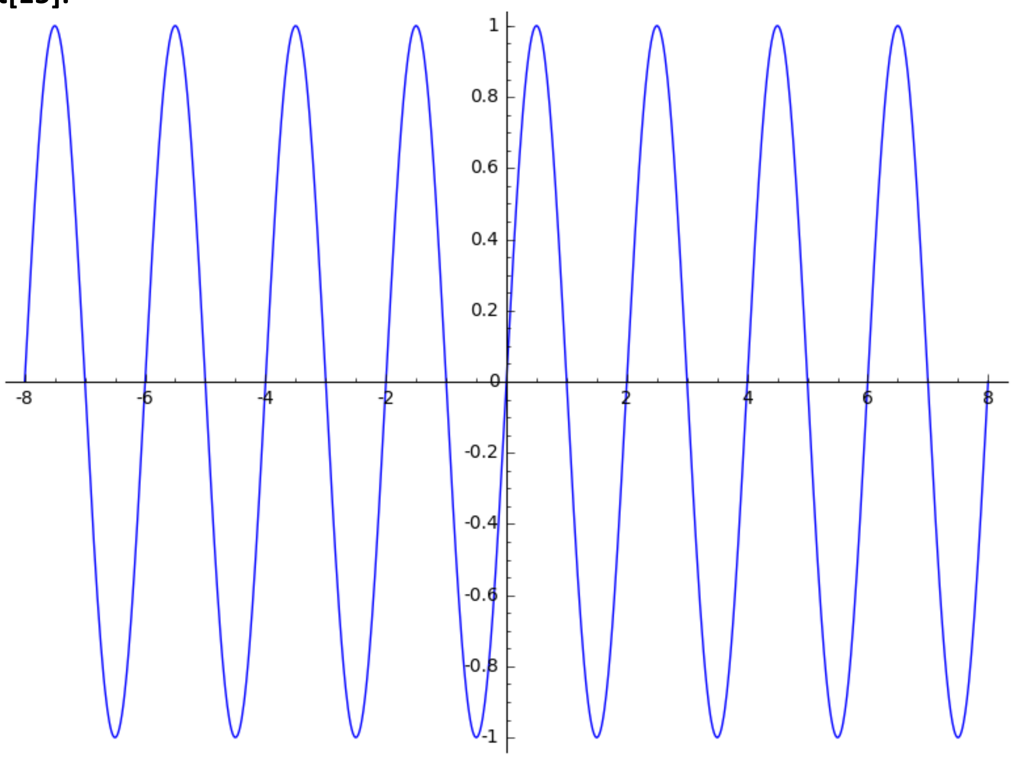
In[14]: plot(sin(pi*x), (x, -8, 8), ticks=2)

Out[14]:



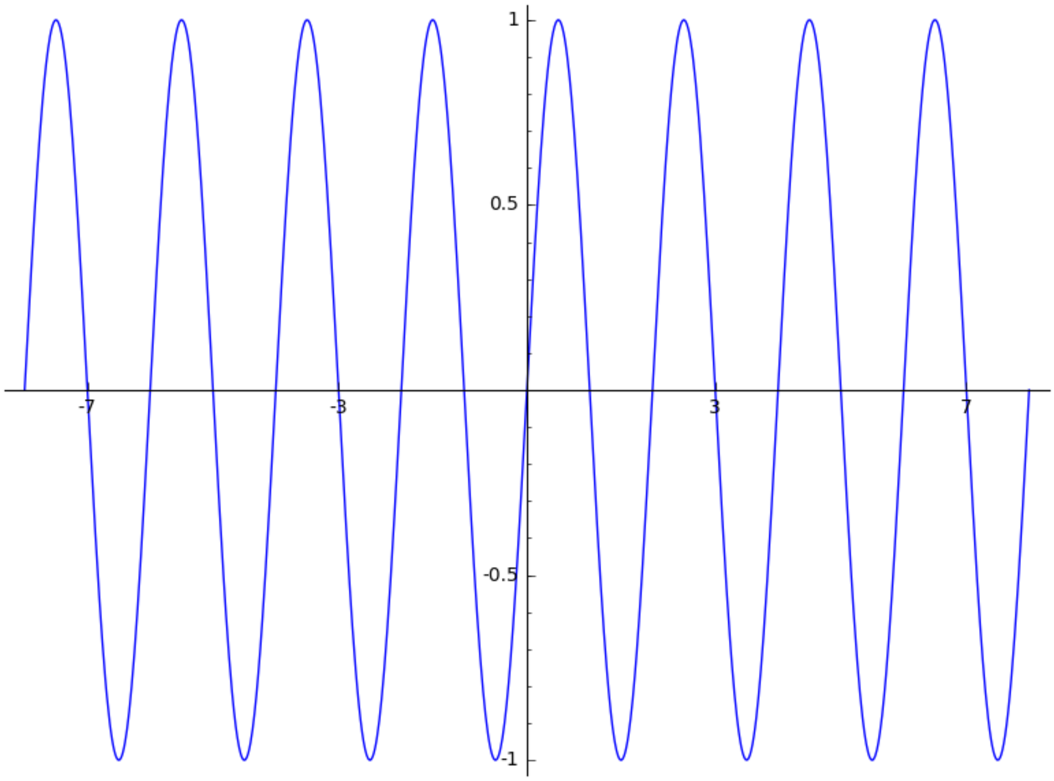
In[15]: plot(sin(pi*x), (x, -8, 8), ticks=[2, 0.2])

Out[15]:



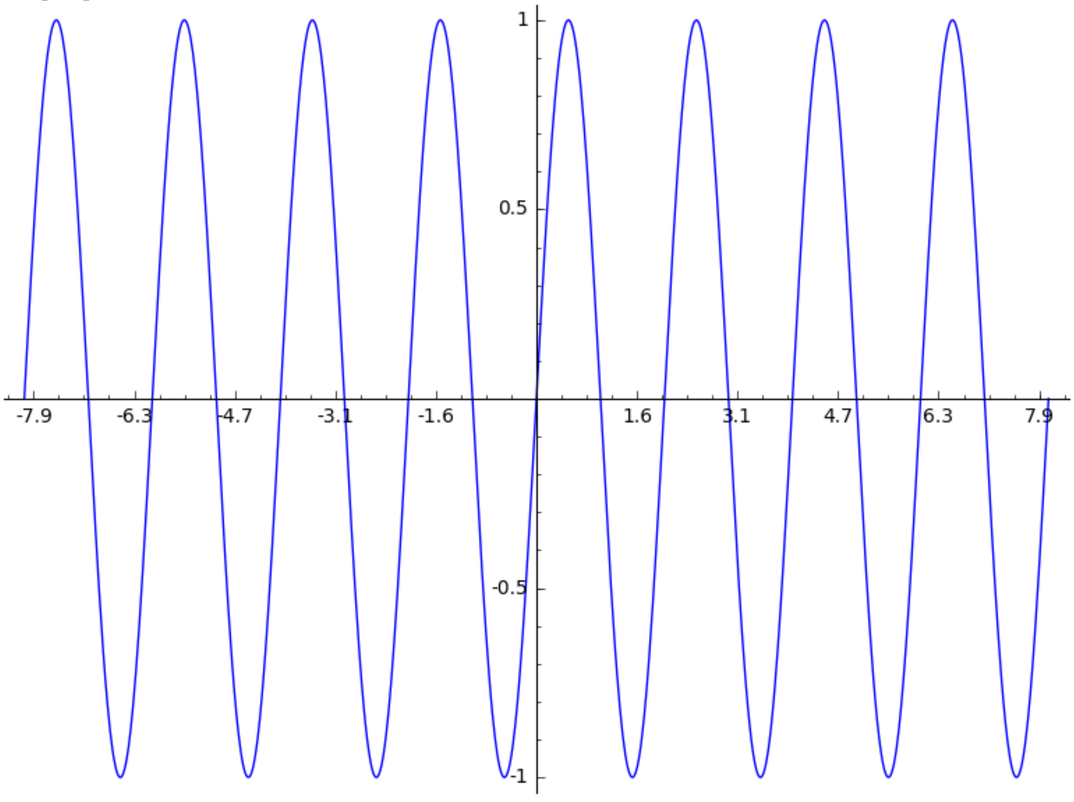
In[16]: plot(sin(pi*x), (x, -8, 8), ticks=[[-7,-3,0,3,7], None])

Out[16]:



In[17]: plot(sin(pi*x), (x, -8, 8), ticks= pi/2)

Out[17]:

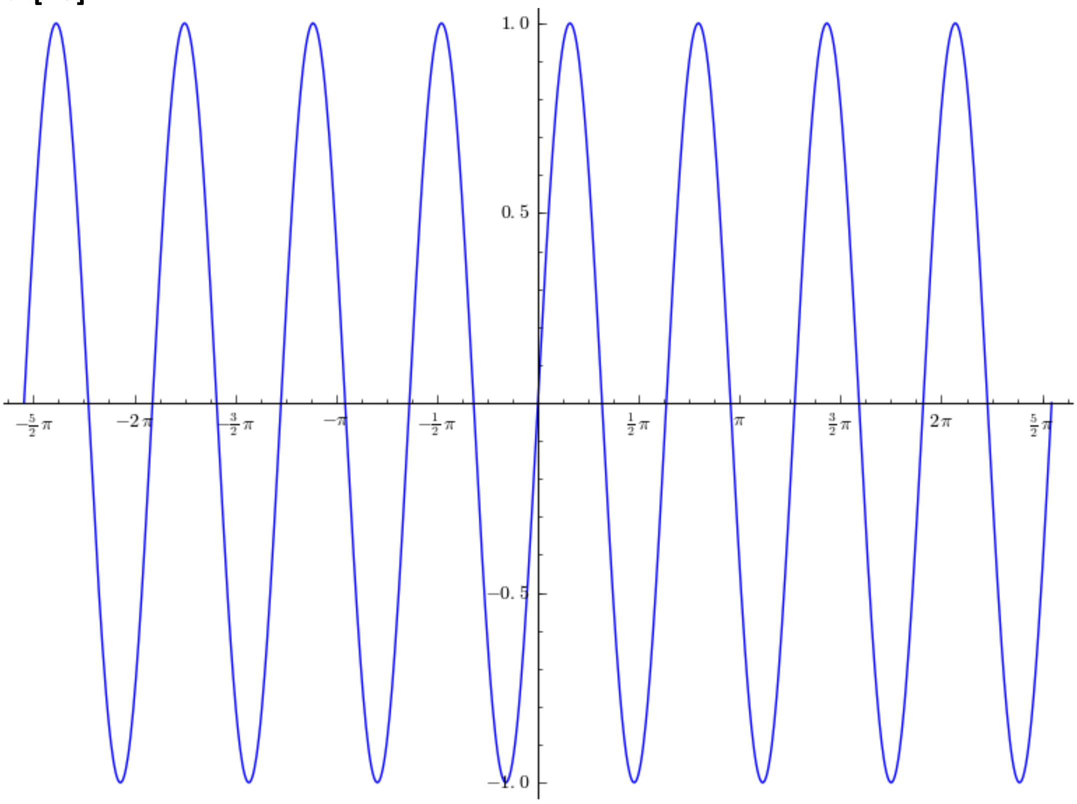


Η επιλογή tick_formatter καθορίζει τη μορφή που θα εμφανίζονται οι τιμές.

tick_formatter = form	οι τιμές εμφανίζονται σύμφωνα με τη μορφή form.

In[18]: plot(sin(pi*x), (x, -8, 8), ticks=pi/2, tick_formatter=pi)

Out[18]:



2.2.13 fill

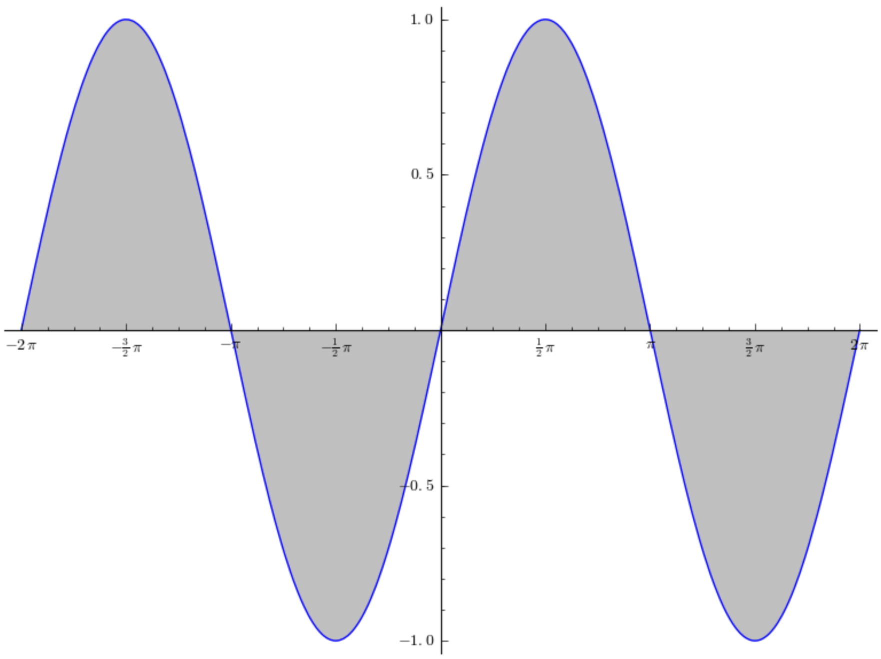
Η επιλογή ***fill*** γεμίζει με χρώμα συγκεκριμένα χωρία της γραφικής παράστασης.

fill = None (προεπιλογή)	δεν έχουμε γέμισμα με χρώμα κάποιου χωρίου της γραφικής παράστασης
fill = ‘axis’	γεμίζει με χρώμα τη περιοχή μεταξύ της γραφικής παράστασης και του άξονα xx’
fill = ‘min’	γεμίζει με χρώμα τη περιοχή μεταξύ της γραφικής παράστασης και της ελάχιστης τιμής της
fill = ‘max’	γεμίζει με χρώμα τη περιοχή μεταξύ της γραφικής παράστασης και της μέγιστης τιμής της

fill = c	όπου c αριθμός, γεμίζει με χρώμα τη περιοχή μεταξύ της γραφικής παράστασης και της ευθείας y = c
fill = g	όπου g συνάρτηση, γεμίζει με χρώμα τη περιοχή μεταξύ της γραφικής παράστασης και της συνάρτησης g.

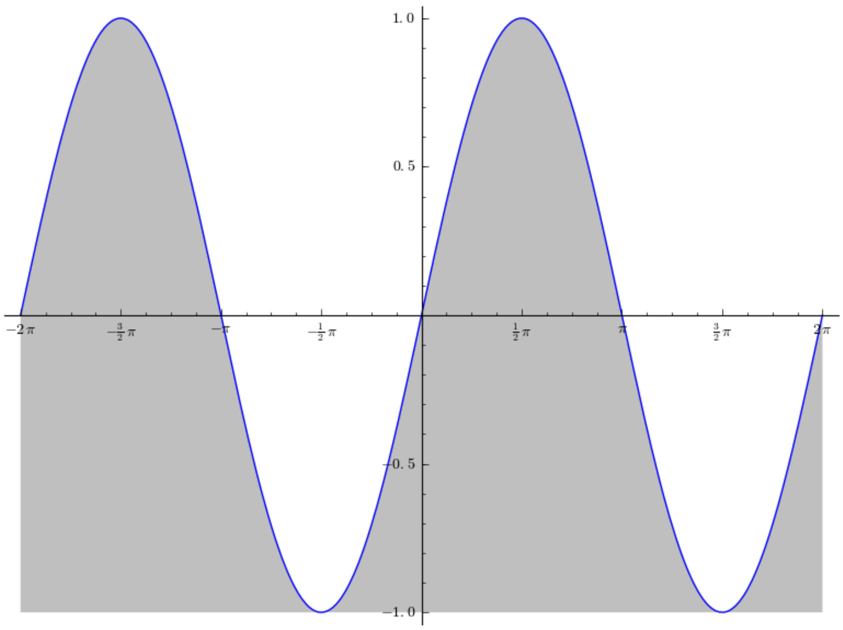
In[19]: plot(sin(x), (x, -2*pi, 2*pi), ticks=pi/2, tick_formatter=pi, fill='axis')

Out[19]:



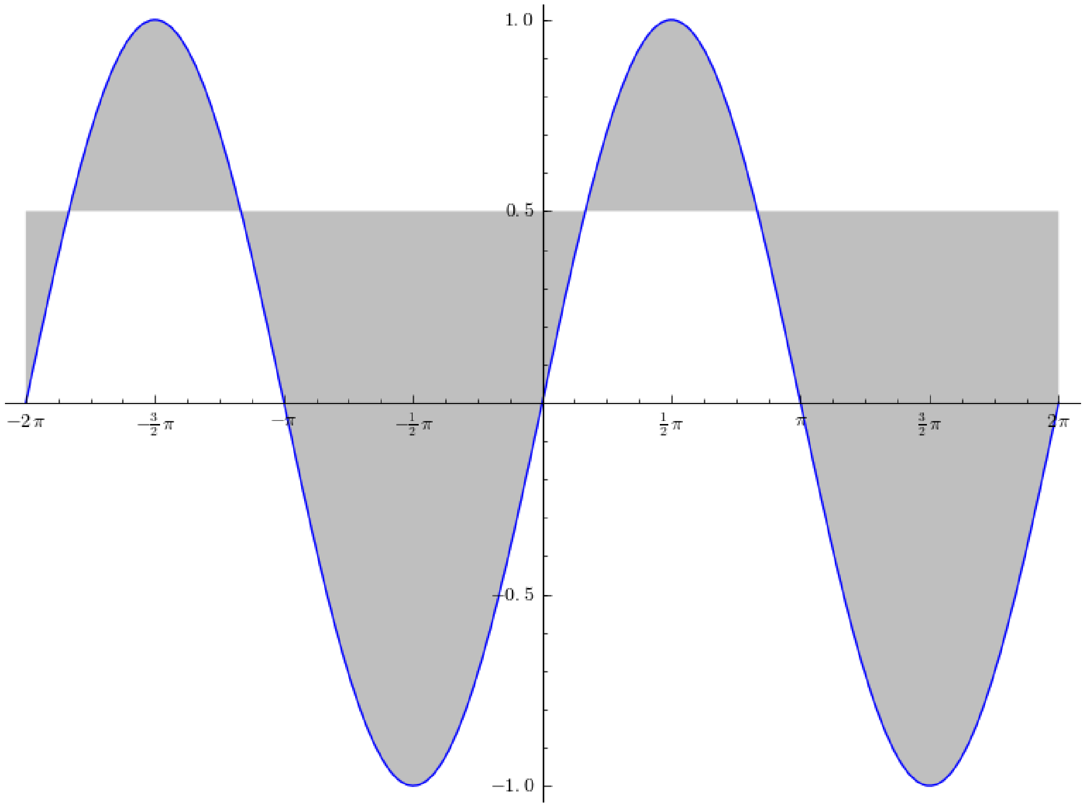
In[20]: plot(sin(x), (x, -2*pi, 2*pi), ticks=pi/2, tick_formatter=pi, fill='min')

Out[20]:



In[21]: plot(sin(x), (x, -2*pi, 2*pi), ticks=pi/2, tick_formatter=pi, fill=0.5)

Out[21]:

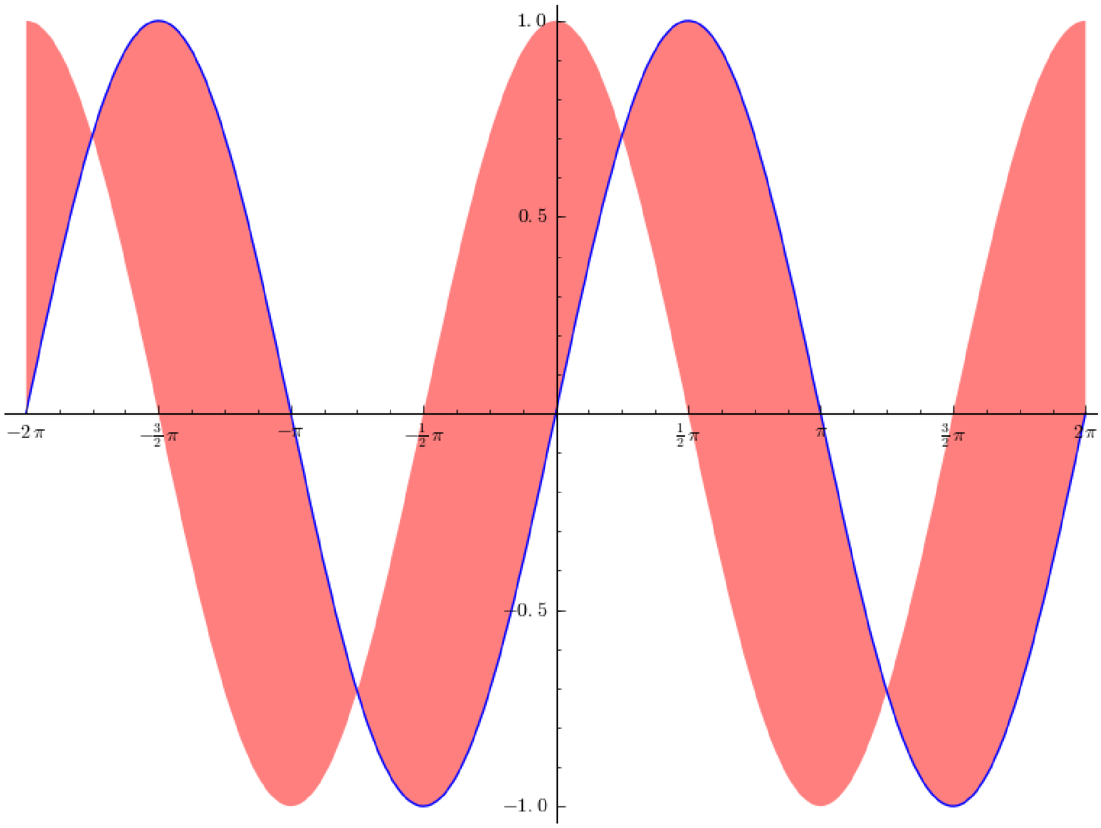


Η επιλογή ***fillcolor*** μας επιτρέπει να επιλέξουμε το χρώμα με το οποίο θα γεμίζουμε χωρία της γραφικής παράστασης.

fillcolor = None (προεπιλογή)	το προκαθορισμένο χρώμα που είναι το γκρι
fillcolor = (r, g, b)	καθορίζει το χρώμα ως συνδυασμός τριών χρωμάτων, του κόκκινου (r), του πράσινου (g) και του μπλε (b). Δίνοντας σε κάθε ένα από τα r, g, b, τιμές στο διάστημα [0, 1]
fillcolor = string	το χρώμα καθορίζεται με τη χρήση μίας συμβολοσειράς π.χ. ‘red’, ‘blue’, ‘purple’ κ.τ.λ.

In[22]: plot(sin(x), (x, -2*pi, 2*pi), ticks=pi/2, tick_formatter=pi, fill=cos(x), fillcolor=(1,0,0))

Out[22]:



2.2.14 Προσθήκη κειμένου στη γραφική παράσταση

Η συνάρτηση ***text*** μας επιτρέπει να προσθέτουμε κείμενο σε μία γραφική παράσταση.

text('κείμενο', (x, y), επιλογές]	εκτυπώνει το κείμενο κεντραρισμένο στο σημείο (x, y) με συγκεκριμένες επιλογές.

Οι πιο χρήσιμες επιλογές της συνάρτησης ***text*** είναι:

fontsize	καθορίζει το μέγεθος των χαρακτήρων. Η τιμή που παίρνει είναι: είτε ένας ακέραιος που καθορίζει το μέγεθος είτε ένα αλφαριθμητικό που αντιστοιχεί σε κάποιο προκαθορισμένο μέγεθος ('xx-small', 'x-small', 'small', 'medium', 'large', 'x-large', 'xx-large').
fontstyle	καθορίζει το στυλ των χαρακτήρων. Η τιμή που παίρνει είναι ένα αλφαριθμητικό που αντιστοιχεί σε κάποιο προκαθορισμένο στυλ ('normal', 'italic', 'oblique').
rgbcolor	καθορίζει το χρώμα των χαρακτήρων ως συνδυασμός τριών χρωμάτων, του κόκκινου (r), του πράσινου (g) και του μπλε (b). Δίνοντας σε κάθε ένα από τα r, g, b, τιμές στο διάστημα [0, 1].

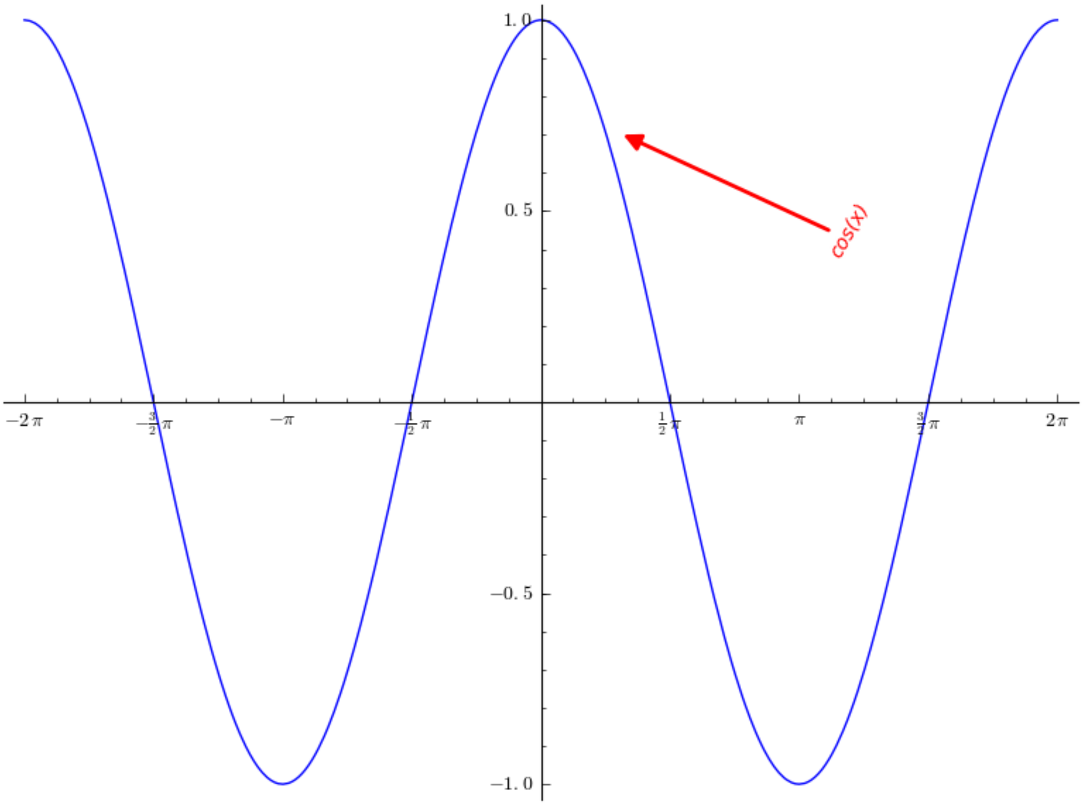
rotation	καθορίζει τη στροφή του κειμένου. Η τιμή του είναι ένας αριθμός που αντιστοιχεί σε μοίρες.

Η συνάρτηση ***arrow*** σχεδιάζει βέλη.

arrow((x1, y1), (x2, y2), επιλογές)	Σχεδιάζει ένα βέλος με αρχή το σημείο (x1, y1) και τέλος το σημείο (x2, y2) με συγκεκριμένες επιλογές (π.χ. rgbcolor).

```
In[23]:= g1=plot(cos(x), (x, -2*pi, 2*pi), ticks=pi/2, tick_formatter=pi)
         g2=arrow((3.5, 0.45) , (1, 0.7 ), rgbcolor='red')
         g3=text('cos(x)', (3.75, 0.45), fontstyle='italic', rgbcolor=(1,0,0),
                  rotation=60)
         g1+g2+g3
```

Out[23]:=



2.3 Επιπλέον συναρτήσεις για δισδιάστατα γραφικά

2.3.1 Γραφική παράσταση καμπύλης που ορίζεται με παραμετρικές εξισώσεις

Μερικές φορές, οι καμπύλες ορίζονται παραμετρικά, για παράδειγμα μπορεί οι συντεταγμένες x και y των σημείων της καμπύλης να ορίζονται ως δύο ανεξάρτητες συναρτήσεις μίας τρίτης μεταβλητής. Με την συνάρτηση ***parametric_plot*** μπορούμε να σχεδιάσουμε τη γραφική παράσταση παραμετρικών καμπυλών.

parametric_plot((x(t), y(t)), (t, tmin, tmax), options)	σχεδιάζει τις παραμετρικές εξισώσεις x(t), y(t) στο διάστημα [tmin, tmax].

Για παράδειγμα, έστω ότι θέλουμε να σχεδιάσουμε την γραφική παράσταση της καμπύλης που δίνεται από τις παραμετρικές εξισώσεις:

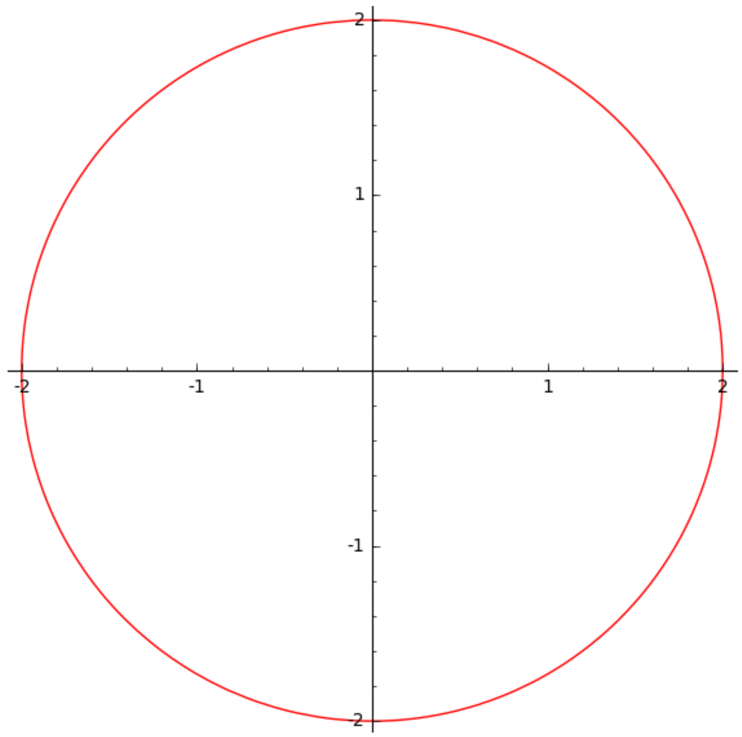
$$x(t)=2cos(t), \quad y(t)=2sin(t), \; t\in[0,2\pi),$$

που αντιστοιχούν στο κύκλο $x^2+y^2=4$.

In[1]: t = var('t')

parametric_plot((2*cos(t), 2*sin(t)), (t, 0, 2*pi), color=(1, 0, 0))

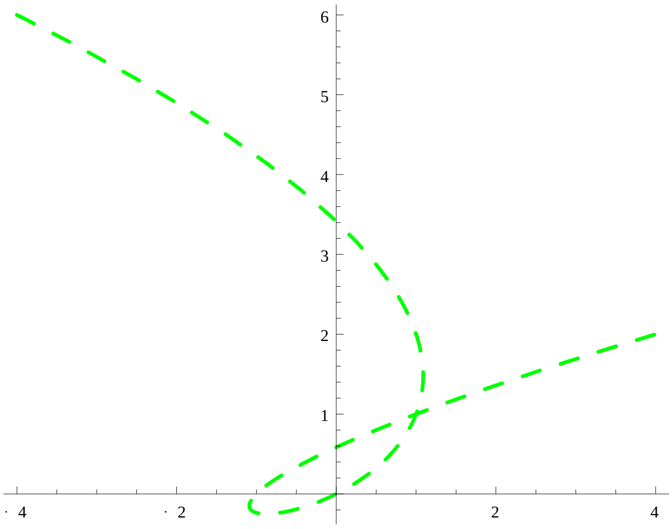
Out[1]:



Παρατηρούμε ότι και στη συνάρτηση ***parametric_plot*** μπορούμε να χρησιμοποιήσουμε τις ίδιες ή παρόμοιες επιλογές που χρησιμοποιήσαμε και στην συνάρτηση ***plot***.

In[2]: parametric_plot((t3 – 2*t, t**2 – t), (t, 0, 2*pi), color='green ', linestyle='--')**

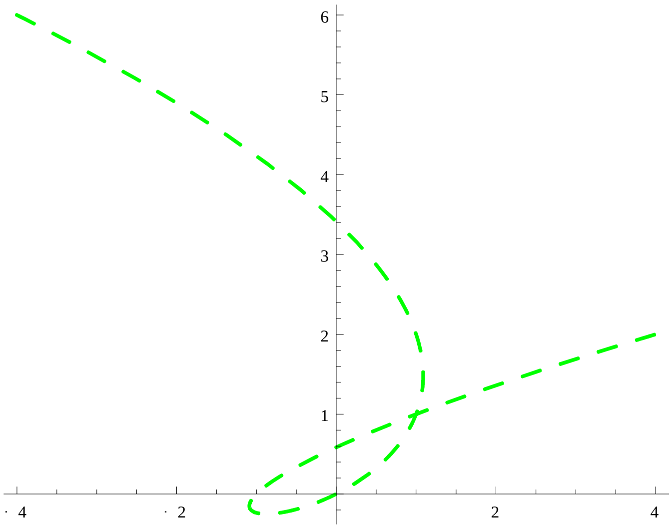
Out[2]:



Η συνάρτηση ***parametric_plot*** είναι ισοδύναμη με τη συνάρτηση ***plot*** όταν στη συνάρτηση ***plot*** χρησιμοποιήσουμε την επιλογή ***parametric = True***.

In[3]: plot((t3 – 2*t, t**2 – t), (t, 0, 2*pi), color='green ', linestyle='--', parametric = True)**

Out[3]:

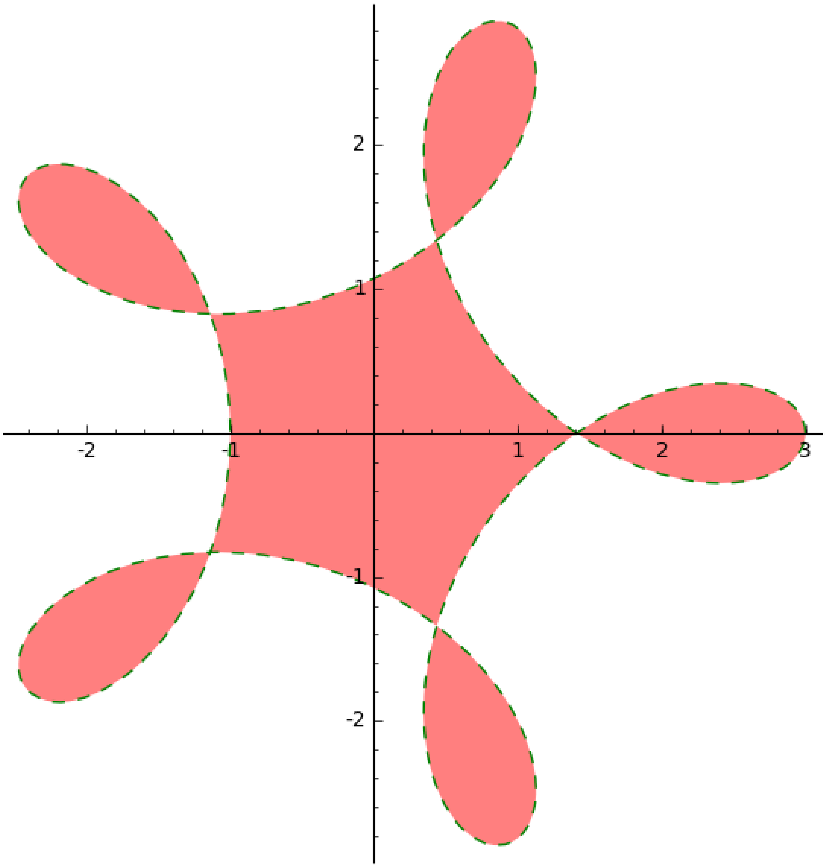


Σημειώστε ότι στη συνάρτηση ***parametric_plot*** η επιλογή ***fill*** έχει μόνο δύο τιμές:

fill = False (προεπιλογή)	δεν έχουμε γέμισμα με χρώμα
fill = True	γεμίζει με χρώμα το χωρίο που ορίζει η συνάρτηση.

In[4]: parametric_plot((cos(t) + 2*cos(t/4), sin(t) – 2*sin(t/4)), (t, 0, 8*pi), color='green ', linestyle='--', fill = True, fillcolor='red')

Out[4]:



2.3.2 Γραφική παράσταση καμπύλης που ορίζεται πεπλεγμένα

Για να σχεδιάσουμε τη γραφική παράσταση καμπύλης που ορίζεται πεπλεγμένα χρησιμοποιούμε τη συνάρτηση ***implicit_plot***.

implicit_plot(f, (x, xmin, xmax), (y, ymin, ymax), options)

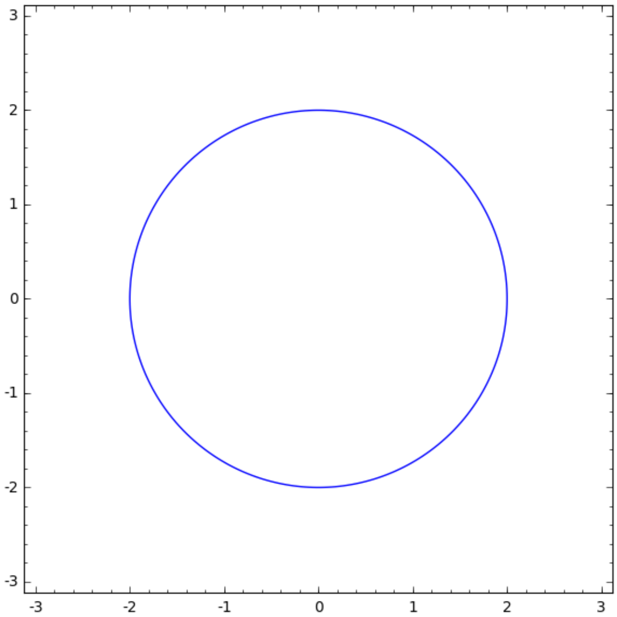
Σχεδιάζει τη γραφική παράσταση της f στα διαστήματα (xmin, xmax) και (ymin, ymax). Όπου f μία συνάρτηση ή μία ισότητα δύο μεταβλητών x και y. Αν η f είναι συνάρτηση, σχεδιάζει τη γραφική παράσταση της f(x, y) = 0.

Προσοχή: Στον ορισμό της ισότητας το ίσον γράφεται ως == (με δύο ίσον).

Ας σχεδιάσουμε τον κύκλο $x^2 + y^2 = 4$ ακτίνας 2 και με κέντρο το σημείο (0, 0).

In[5]: **y = var ('y')**
implictit_plot (x2 + y**2 == 4, (x, -3, 3), (y, -3, 3))**

Out[5]:

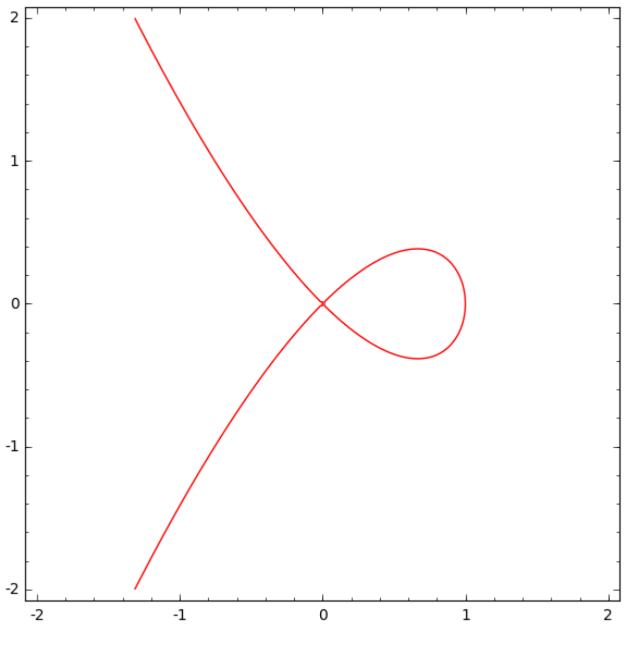


Στη συνάρτηση **implicit_plot** μπορούμε να χρησιμοποιήσουμε επίσης τις ίδιες ή παρόμοιες επιλογές που χρησιμοποιήσαμε και στην συνάρτηση **plot**.

Στη συνέχεια θα σχεδιάσουμε την ελλειπτική καμπύλη $x^2 - y^2 = x^3$ για $x, y \in [-2, 2]$ με κόκκινο χρώμα.

In[6]: implictit_plot (x2 – y**2 == x**3, (x, -2, 2), (y, -2, 2), color = 'red')**

Out[6]:

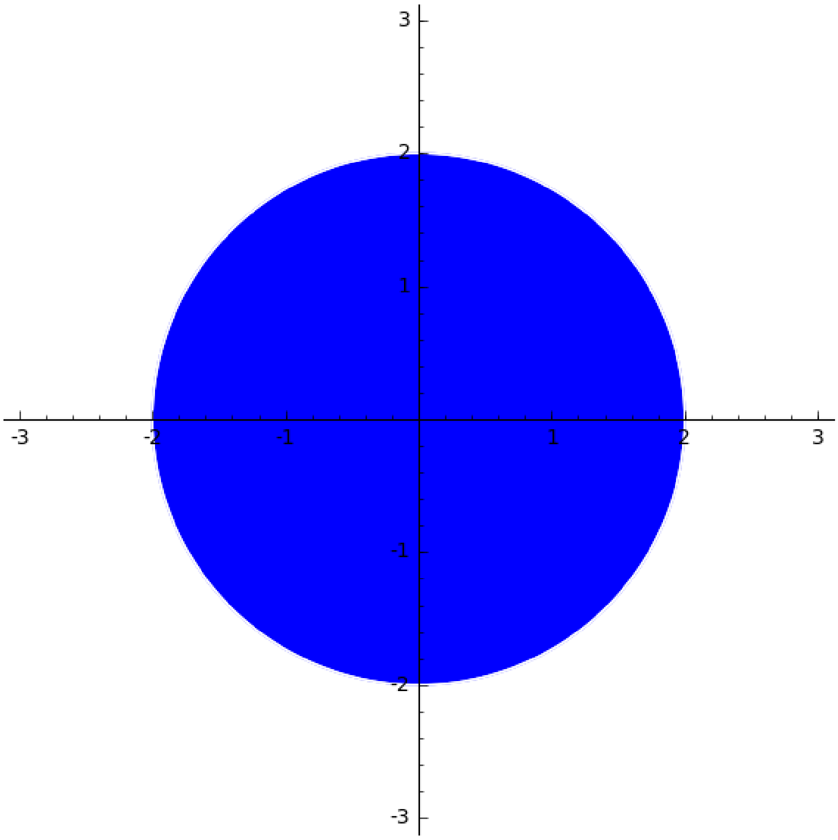


Σημειώστε ότι στη συνάρτηση ***implicit_plot***, όπως και στη συνάρτηση ***parametric_plot***, η επιλογή ***fill*** έχει μόνο δύο τιμές και δεν συνδυάζεται εύκολα με άλλες επιλογές:

fill = False (προεπιλογή)	δεν έχουμε γέμισμα με χρώμα
fill = True	γεμίζει με χρώμα το χωρίο που ορίζει η συνάρτηση.

In[7]:
```
f(x, y) = x**2 + y**2 - 4
implictit_plot (f, (x, -3, 3), (y, -3, 3), fill=True)
```

Out[7]:

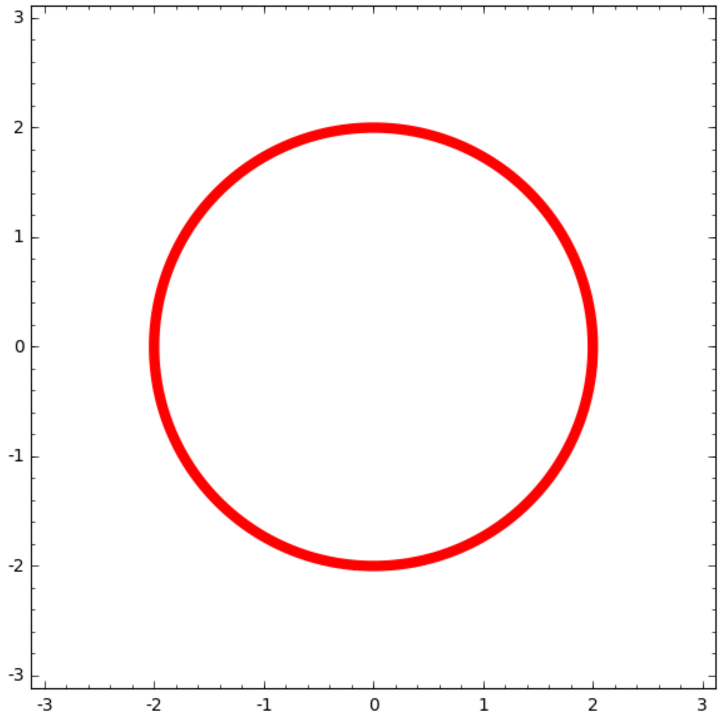


Το πάχος της γραμμής στην συνάρτηση ***implicit_plot*** το καθορίζει η επιλογή ***linewidth*** αντί της επιλογής ***thickness*** που χρησιμοποιείται στη συνάρτηση ***plot***.

linewidth = None (προεπιλογή)	σχεδιάζει γραμμή με το προκαθορισμένο πάχος
linewidth = n	όπου n ακέραιος. Σχεδιάζει γραμμή n φορές παχύτερη.

In[8]: implictit_plot (x2 + y**2 == 4, (x, -3, 3), (y, -3, 3), color='red', linewidth=6)**

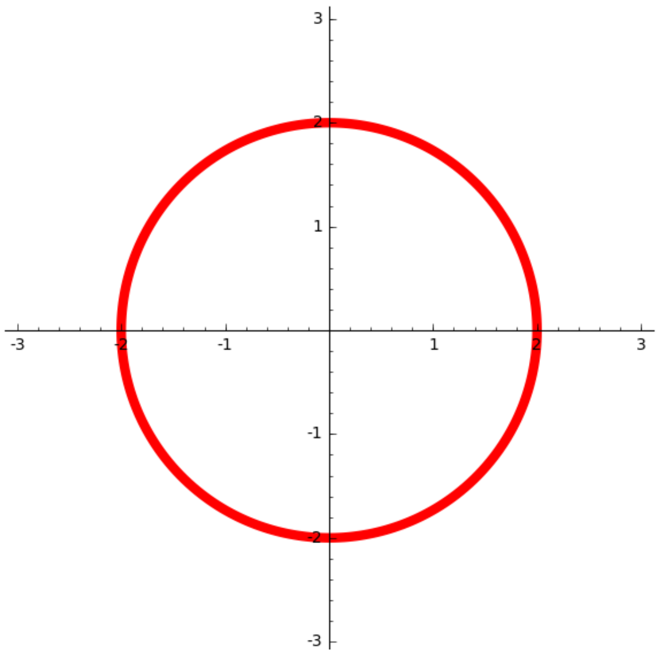
Out[8]:



Αν θέλουμε να εμφανίσουμε τους άξονες και να αφαιρέσουμε το πλαίσιο, τότε θα πρέπει να χρησιμοποιήσουμε τις επιλογές ***axes*** και ***frame***.

In[9]: implictit_plot (x2 + y**2 == 4, (x, -3, 3), (y, -3, 3), color='red', linewidth=6, axes=True, frame=False)**

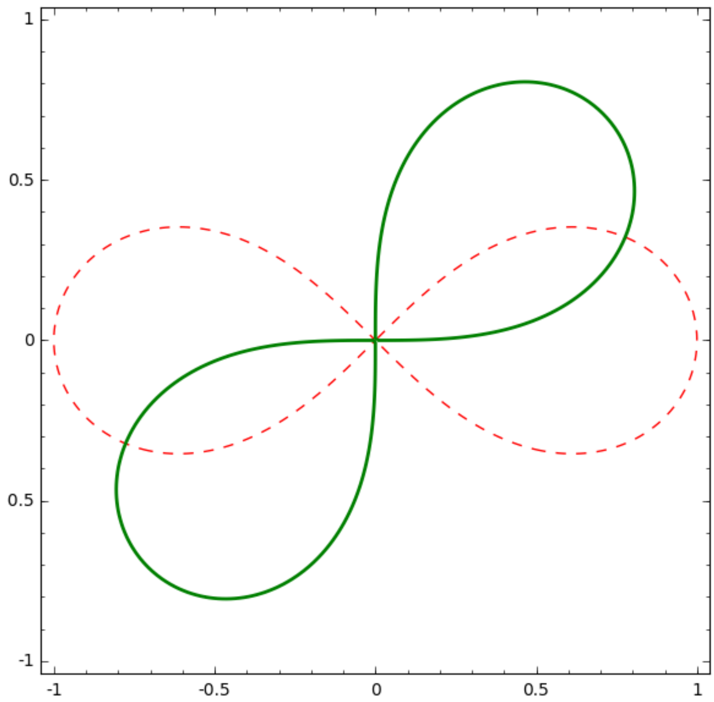
Out[9]:



Στη συνέχεια. ας δούμε και ένα παράδειγμα σχεδιασμού των γραφικών παραστάσεων δύο πεπλεγμένων συναρτήσεων. Των f(x, y) = $\left(x^2+y^2\right)^2 = x^2 - y^2$ και g(x, y) = $\left(x^2+y^2\right)^2 = 2xy$ για $x, y \in [-1,1]$.

In[10]: g1 = implicit_plot((x2 + y**2)**2 == x**2 – y**2, (x, -1, 1), (y, -1, 1), color = 'red', linestyle = '--')**
g2 = implicit_plot((x2 + y**2)**2 == 2*x*y, (x, -1, 1), (y, -1, 1), color = 'green', linewidth=2)**
g1 + g2

Out[10]:



2.3.3 *Γραφική παράσταση καμπύλης που ορίζεται με πολικές συντεταγμένες*

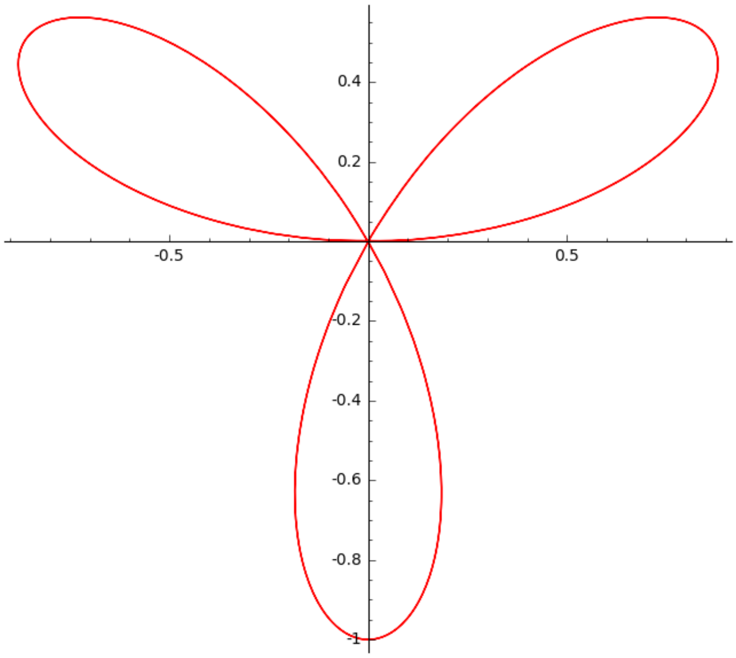
Η γραφική παράσταση μιας καμπύλης που ορίζεται σε πολικές συντεταγμένες, επιτυγχάνεται με τη χρήση της συνάρτησης ***polar_plot***.

polar_plot(f(θ), (θ, θmin, θmax), options)	σχεδιάζει τη γραφική παράσταση της εξίσωσης, υπό μορφή πολικών συντεταγμένων, $r = f(\theta)$, καθώς η γωνία θ μεταβάλλεται από θmin έως θmax
polar_plot((f_1(θ), f_2(θ), …), (θ, θmin, θmax))	σχεδιάζει πολλά πολικά γραφήματα στο ίδιο σύστημα αξόνων.

Επίσης, στη συνάρτηση ***polar_plot*** μπορούμε να χρησιμοποιήσουμε τις ίδιες επιλογές που χρησιμοποιήσαμε και στην συνάρτηση ***plot***.

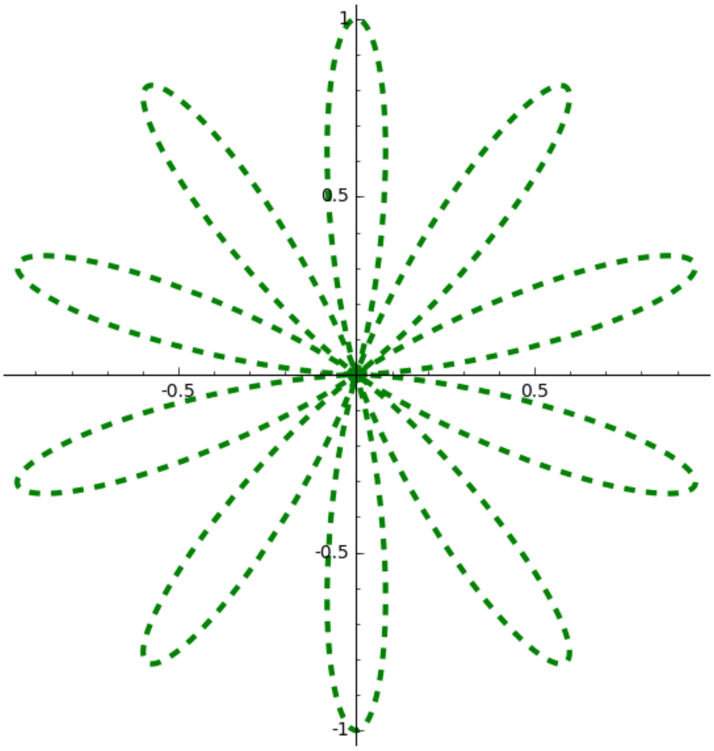
In[11]: polar_plot(sin(3*t), (t, 0, 2*pi), color = 'red')

Out[11]:



In[12]: polar_plot(sin(5*t)^2, (t, 0, 2*pi), color = 'green', linestyle='--', thickness=3)

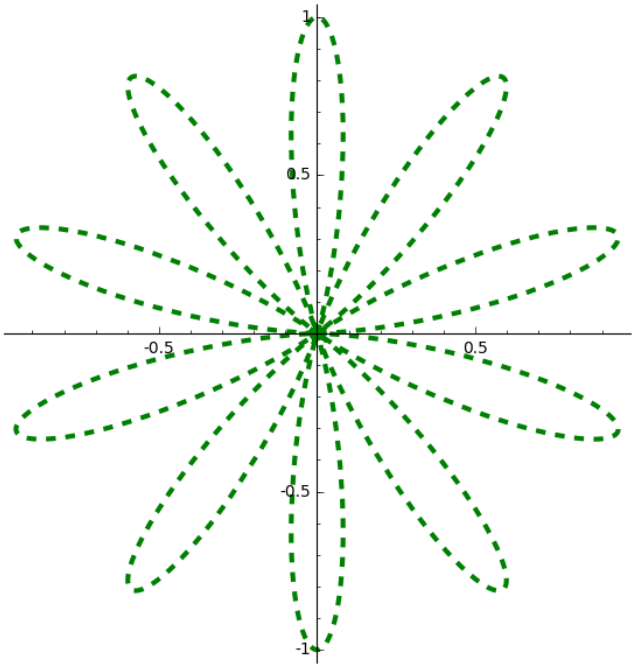
Out[12]:



Η συνάρτηση ***polar_plot*** είναι ισοδύναμη με τη συνάρτηση ***plot*** όταν στη συνάρτηση ***plot*** χρησιμοποιήσουμε τις επιλογές ***polar = True*** και ***aspect_ratio=1***.

In[13]: plot(sin(5*t)^2, (t, 0, 2*pi), color = 'green', linestyle='--', thickness=3, polar=True, aspect_ratio=1)

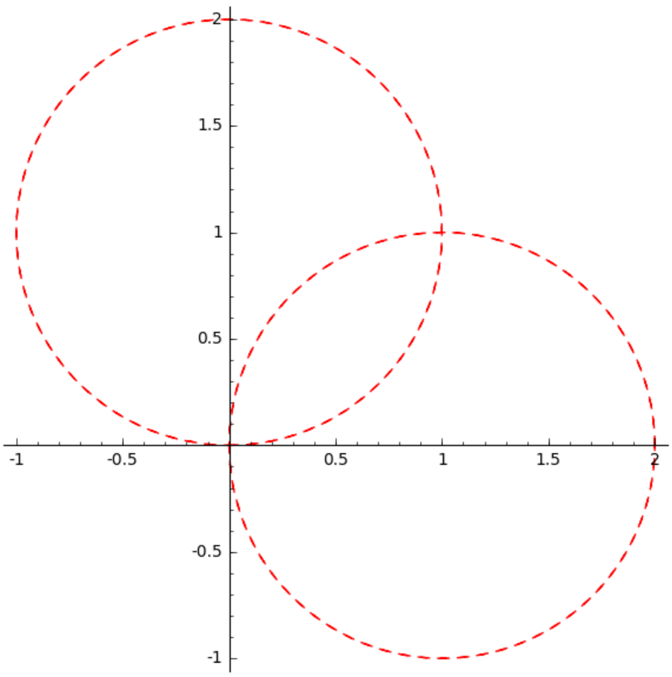
Out[13]:



Μπορούμε να σχεδιάσουμε τη γραφική παράσταση πολλών συναρτήσεων που δίνονται υπό μορφή πολικών συντεταγμένων.

In[14]: polar_plot((2*sin(t), 2*cos(t)), (t, 0, 2*pi), color = 'red', linestyle='--')

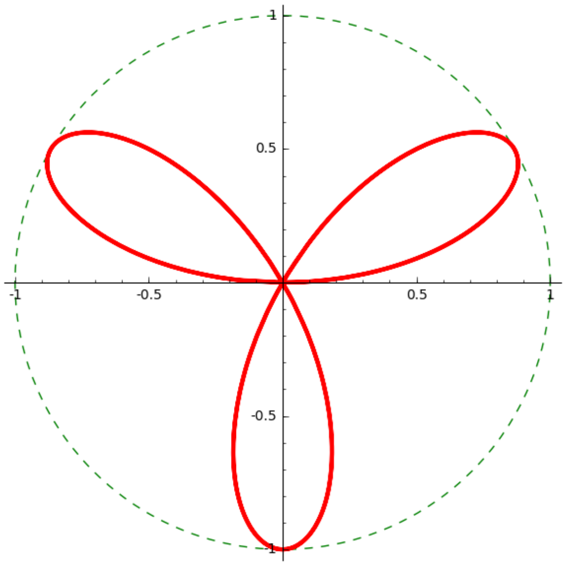
Out[14]:



Παρατηρούμε ότι οι γραφικές παραστάσεις των συναρτήσεων σχεδιάζονται με τις ίδιες επιλογές. Αν θέλουμε να σχεδιάσουμε τις γραφικές παραστάσεις των συναρτήσεων με διαφορετικές επιλογές θα πρέπει αρχικά να τις σχεδιάσουμε ξεχωριστά και μετά να τις εμφανίσουμε στο ίδιο σύστημα αξόνων.

In[15]:
```
g1=polar_plot(1, (t, 0, 2*pi), color='green', linestyle='--')
g2=polar_plot(sin(3*t), (t, 0, 2*pi), color='red', thickness=3)
g1+g2
```

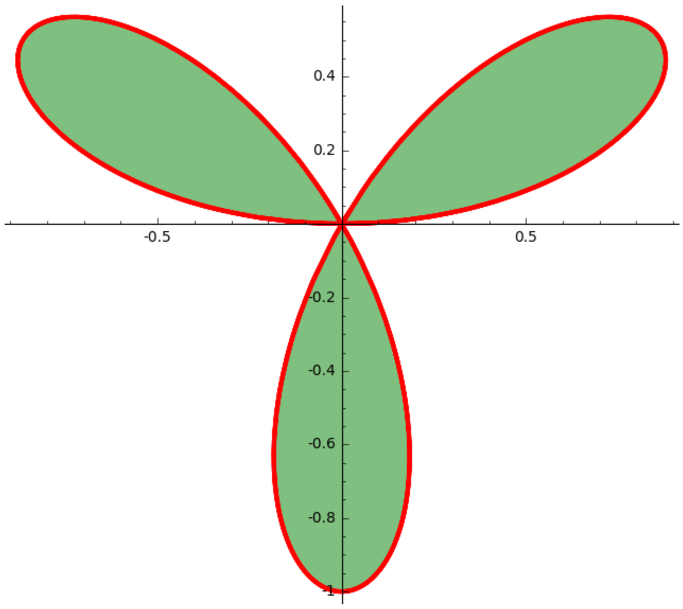
Out[15]:



Μπορούμε με τις επιλογές ***fill*** και ***fillcolor*** να χρωματίσουμε με συγκεκριμένο χρώμα το χωρίο που ορίζει η γραμμή (καμπύλη) της συνάρτησης.

In[16]:
```
polar_plot(sin(3*t), (t, 0, 2*pi), color='red', thickness=3, fill=True,
                                        fillcolor='green')
```

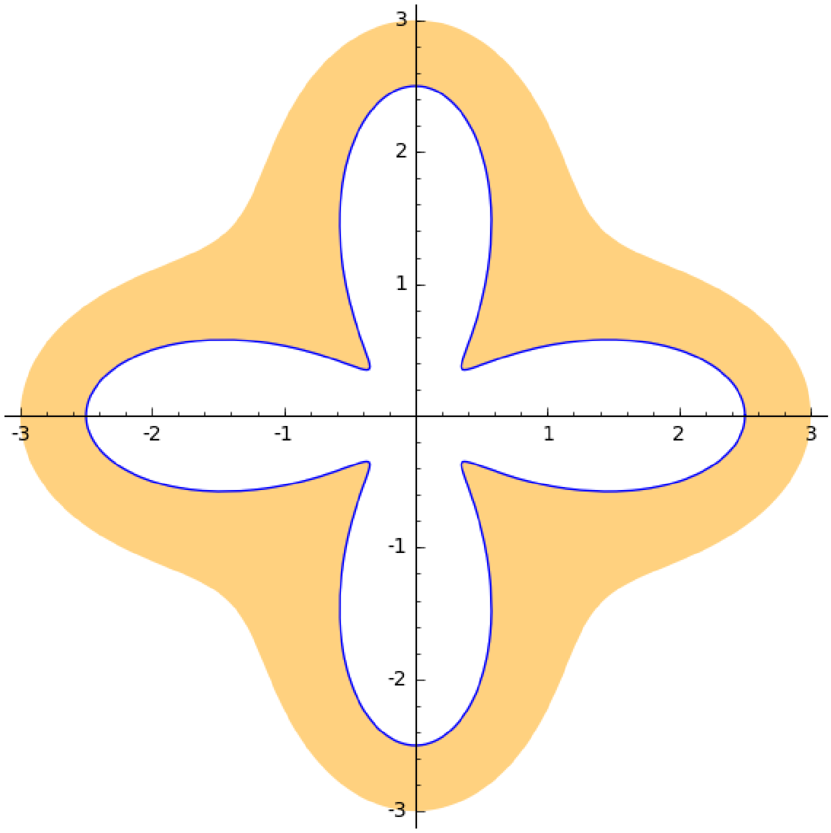
Out[16]:



Επίσης μπορούμε να χρωματίσουμε με συγκεκριμένο χρώμα το χωρίο που ορίζει μεταξύ δύο συναρτήσεων που ορίζονται με πολικές συντεταγμένες.

In[17]: polar_plot(cos(4*t) + 1.5, (t, 0, 2*pi), fill=0.5*cos(4*t) + 2.5, fillcolor='orange')

Out[17]:



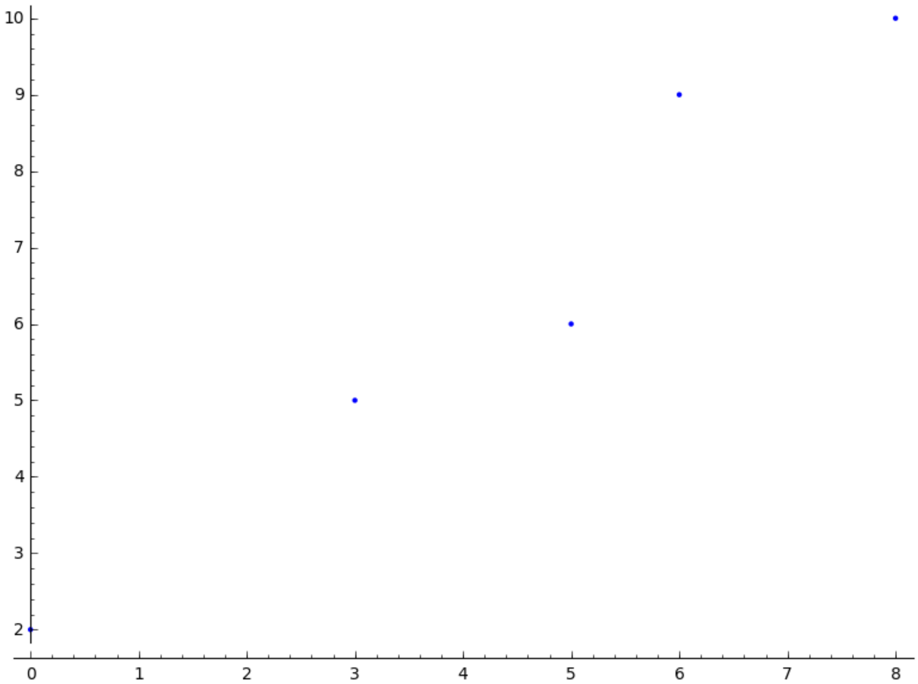
2.3.4 Γραφική παράσταση διακριτών συναρτήσεων

Για να σχεδιάσουμε τη γραφική παράσταση διακριτών συναρτήσεων, δηλαδή, συναρτήσεων που ορίζονται σε ένα διακριτό σύνολο, μπορούμε να χρησιμοποιήσουμε την συνάρτηση ***list_plot***.

list_plot((y_1, y_2, …), options)	σχεδιάζει τα σημεία με συντεταγμένες y: y_1, y_2, … ενώ ως συντεταγμένες x λαμβάνονται οι ακέραιοι 0, 1, 2, …
list_plot(((x_1, y_1), (x_2, y_2), …), options)	σχεδιάζει τα σημεία (x_1, y_1), (x_2, y_2) ,…

In[18]: list_plot(((0,2), (3,5), (5,6), (6,9), (8,10)))

Out[18]:

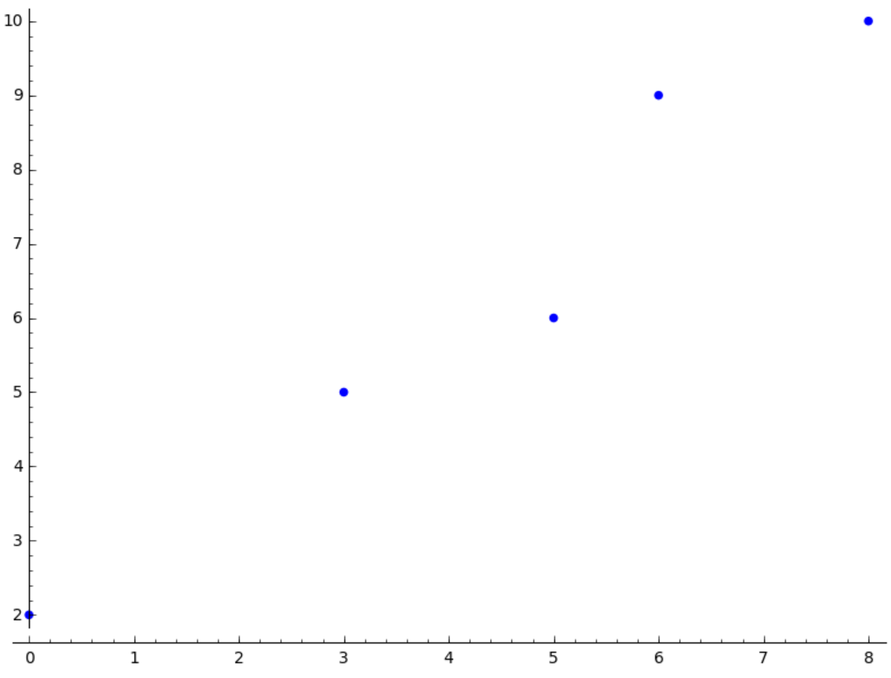


Στη συνάρτηση ***list_plot*** μπορούμε να χρησιμοποιήσουμε διάφορες επιλογές. Οι πιο χρήσιμες επιλογές είναι οι εξής:

plotsize = n	όπου n ακέραιος. Καθορίζει το μέγεθος των σημείων. Η προεπιλεγμένη τιμή είναι 10.

In[19]: list_plot(((0,2), (3,5), (5,6), (6,9), (8,10)), pointsize=30)

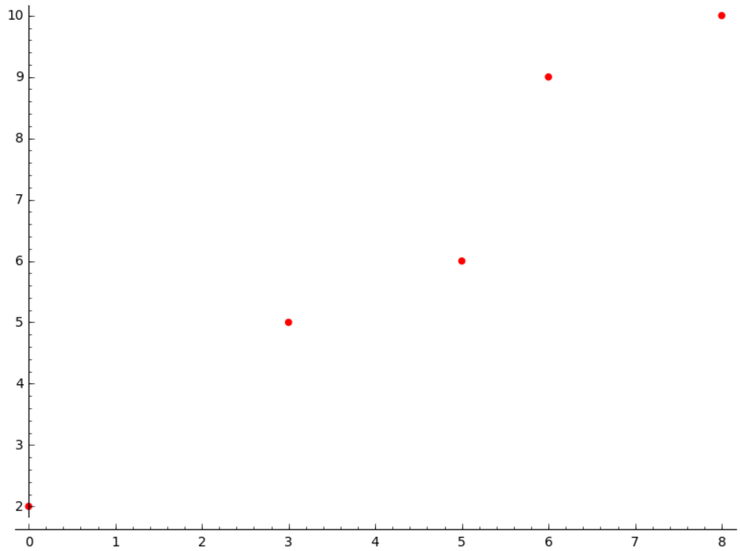
Out[19]:



color = (r, g, b)	καθορίζει το χρώμα ως συνδυασμός τριών χρωμάτων, του κόκκινου (r), του πράσινου (g) και του μπλε (b). Δίνοντας σε κάθε ένα από τα r, g, b, τιμές στο διάστημα [0, 1]. Εξ' ορισμού έχει την τιμή (0, 0, 1) που αντιστοιχεί στο μπλε χρώμα.
color = string	το χρώμα καθορίζεται με τη χρήση μίας συμβολοσειράς π.χ. 'red', 'blue', 'purple' κ.τ.λ.

In[20]: list_plot(((0,2), (3,5), (5,6), (6,9), (8,10)), pointsize=30, color='red')

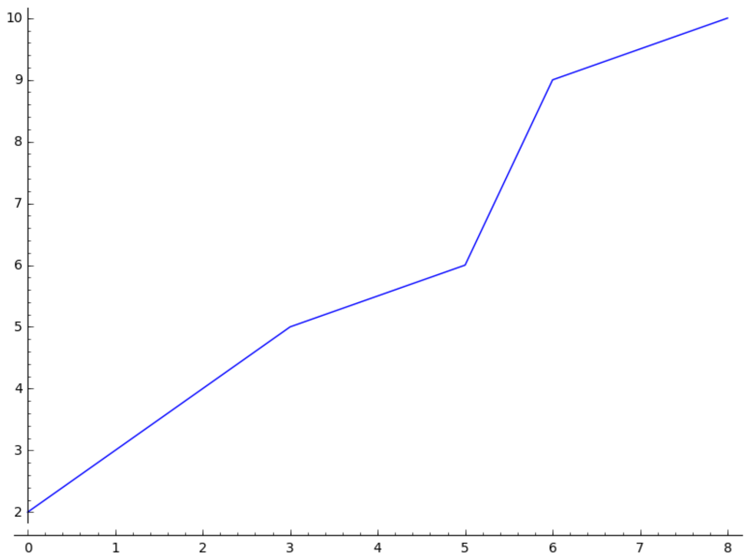
Out[20]:



plotjoined = True	συνδέει τα σημεία με μία καμπύλη
plotjoined = False (προεπιλογή)	δεν συνδέει τα σημεία με μία καμπύλη.

In[20]: list_plot(((0,2), (3,5), (5,6), (6,9), (8,10)), plotjoined=True)

Out[20]:

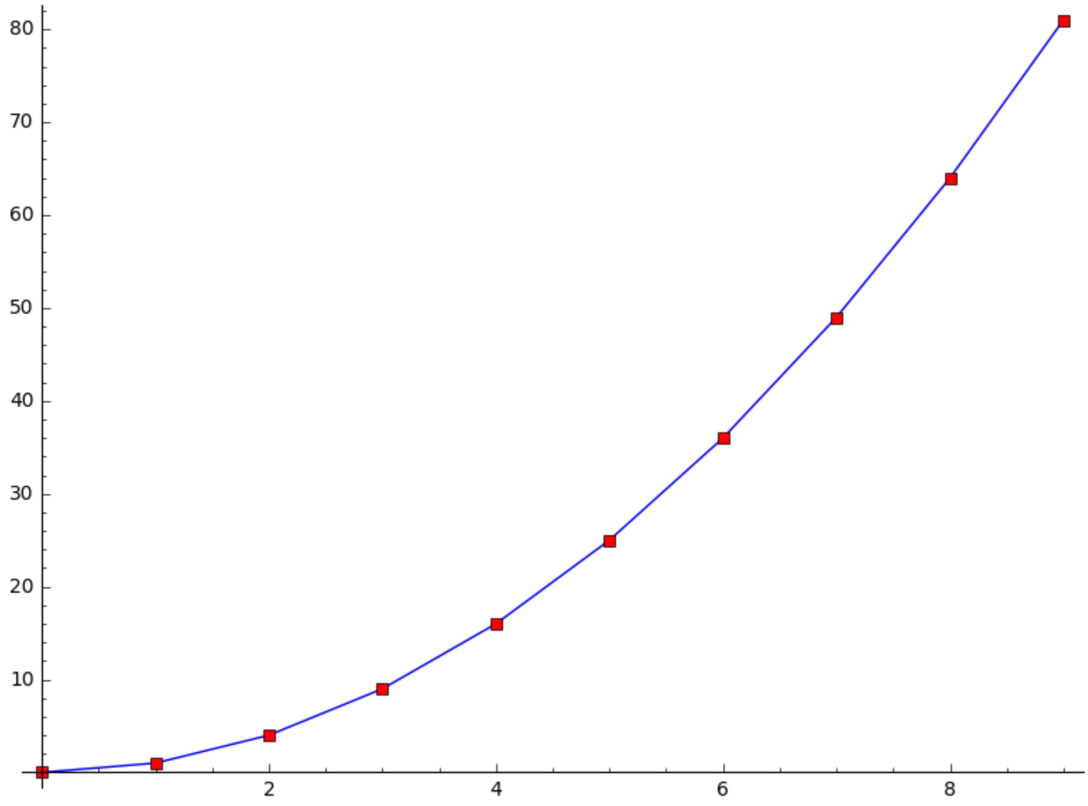


Αν θέλουμε να εμφανίζονται και τα σημεία τότε πρέπει να χρησιμοποιήσουμε τις παρακάτω επιλογές:

marker = 'symbol'	καθορίζει το σχήμα των σημείων: 'o' (κύκλος), 'p' (πεντάγωνο), 's' (τετράγωνο), 'x' (x), '+' (συν), '*' (αστέρι) κ.τ.λ.
markersize = n	καθορίζει το μέγεθος των σημείων.
markerfacecolor= (r, g, b) ή ‘string’	καθορίζει το χρώμα των σημείων.

In[21: **l_s=[k^2 for k in range(10)]**
list_plot(l_s, plotjoined=True, marker='s', markerfacecolor=(1, 0 ,0))

Out[21]:=



3. Τρισδιάστατα γραφικά

3.1 Τρισδιάστατες γραφικές παραστάσεις συναρτήσεων δύο μεταβλητών.

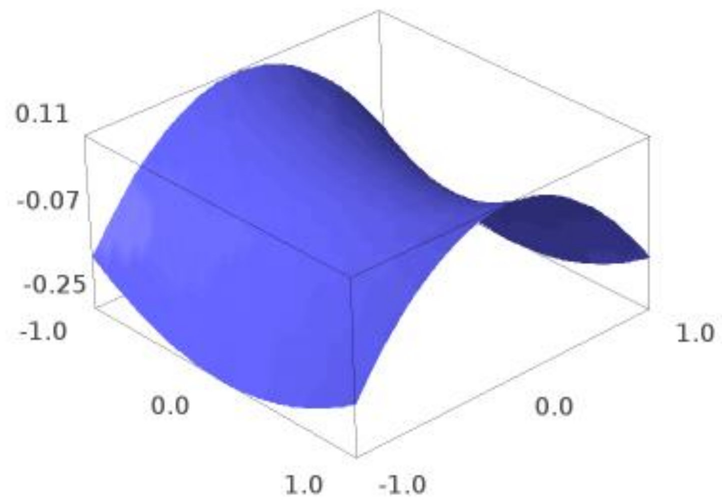
Μία συνάρτηση δύο μεταβλητών μπορεί να θεωρηθεί ως μία τρισδιάστατη επιφάνεια. Η βασική εντολή σχεδίασης, του Sage, μιας γραφικής παράστασης μίας συνάρτησης δύο μεταβλητών είναι η συνάρτηση ***plot3d***.

plot3d(f(x, y), (x, xmin, xmax), (y, ymin, ymax), options)	
	σχεδιάζει τη γραφική παράσταση της συνάρτησης f(x, y) πάνω στο ορθογώνιο που ορίζουν τα διαστήματα [xmin, xmax] και [ymin, ymax].

Ας σχεδιάσουμε τη γραφική παράσταση της συνάρτησης f(x, y) = $\frac{x^2}{9}-\frac{y^2}{4}$ για $x, y \in [-1,1]$.

In[1]: y=var('y')
plot3d(x^2/9 - y^2/4, (x, -1, 1), (y, -1, 1))

Out[1]:=



Η επιφάνεια αυτή δεν είναι άλλη από το υπερβολικό παραβολοειδές δηλ. το «σαμάρι».

Σημειώστε ότι αν πάμε τον κέρσορα του ποντικιού πάνω στη σχεδιασμένη επιφάνεια και κρατώντας πατημένο το αριστερό κουμπί του ποντικιού και ταυτόχρονα κινώντας το ποντίκι μπορούμε να δούμε από διάφορες όψεις τη σχεδιασμένη επιφάνεια.

3.2 Οι επιλογές των τρισδιάστατων γραφικών

Το Sage, προκειμένου να αποδώσει καλύτερα τη τρισδιάστατη γραφική παράσταση με αισθητικά ικανοποιητικό τρόπο, ρυθμίζει από μόνο του όλες τις λεπτομέρειες που θα οδηγήσουν στο αποτέλεσμα αυτό (π.χ. κλίμακα, τοποθέτηση αξόνων, χρωματισμούς, κ.τ.λ.) και τις περισσότερες φορές το επιτυγχάνει.

Παρ' όλα αυτά το Sage μας δίνει την δυνατότητα να επηρεάσουμε το αποτέλεσμα με διάφορους τρόπους, ώστε να πάρουμε τη τρισδιάστατη γραφική παράσταση που επιθυμούμε. Αυτό γίνεται παραθέτοντας στη σύνταξη της εντολής ***plot3d***, οποιεσδήποτε από τις επιλογές που τη συνοδεύουν, και μάλιστα γράφοντας τις με οποιαδήποτε σειρά.

Κάθε επιλογή έχει μία προκαθορισμένη τιμή (προεπιλογή), την οποία παίρνει αυτόματα από το Sage εφόσον δεν ζητήσουμε κάτι διαφορετικό.

Υπάρχει αρκετή ομοιότητα στις επιλογές για την γραφική παράσταση επιφανειών και καμπυλών στο χώρο, με τις αντίστοιχες που γνωρίσαμε στα δισδιάστατα γραφικά. Στη συνέχεια θα εξετάσουμε μερικές από τις επιλογές που χρησιμοποιούνται συχνότερα.

3.2.1 adaptive

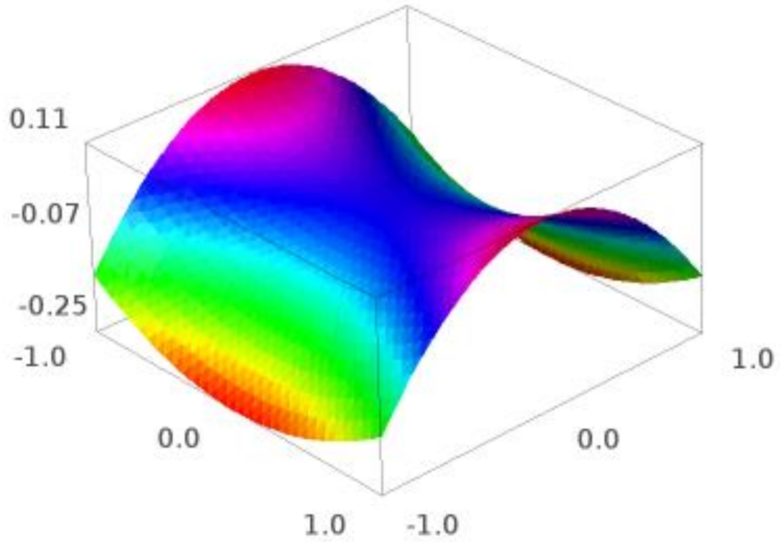
Παρατηρούμε, ότι η επιφάνεια σχεδιάζεται μονόχρωμα. Με την επιλογή ***adaptive*** μπορούμε να σχεδιάσουμε με φινέτσα την επιφάνεια χρησιμοποιώντας περισσότερα χρώματα.

adaptive = False (προεπιλογή)	σχεδιάζει την επιφάνεια μονόχρωμα.
adaptive = True	σχεδιάζει με φινέτσα την επιφάνεια, χρησιμοποιώντας περισσότερα χρώματα.

Σημειώστε ότι η χρήση της επιλογής adaptive οδηγεί σε ποιο αργή αλλά πιο φινετσάτη σχεδίαση. Επίσης δεν

In[2]: plot3d(x^2/9 - y^2/4, (x, -1, 1), (y, -1, 1), adaptive=True)

Out[2]:=



3.2.2 plot_points

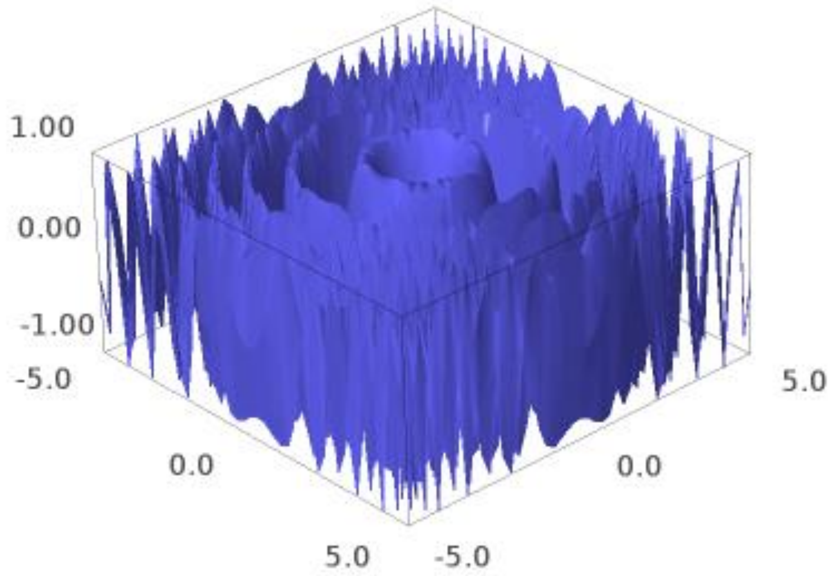
Για να σχεδιαστεί η επιφάνεια επιλέγονται (αυτόματα) ένας αριθμός σημείων σε κάθε κατεύθυνση, n_x και n_y αντίστοιχα, οπότε έχουμε n_x x n_y σημεία πάνω στο επίπεδο Oxy. Αυτά είναι τα δειγματοληπτικά σημεία (plot_points) δηλ. σε κάθε ένα από αυτά υπολογίζεται η τιμή της συνάρτησης f(x, y) και στην συνέχεια με βάση αυτές τις τιμές, σχεδιάζεται η επιφάνεια .

Η επιλογή ***plot_points*** καθορίζει τα σημεία που χρησιμοποιεί το Sage σε κάθε άξονα για τη δημιουργία των γραφημάτων. Όπως είδαμε η προεπιλογή είναι αυτόματη. Αυτό έχει ως αποτέλεσμα πολλές φορές τη δημιουργία γραφημάτων με «οδοντωτές» επιφάνειες. Είναι λοιπόν ευνόητο ότι στις περιπτώσεις που θέλουμε καλύτερη ακρίβεια στην γραφική παράσταση να παίρνουμε περισσότερα σημεία (plot_points).

plot_points = ‘automatic’ (προεπιλογή)	σε κάθε άξονα επιλέγονται αυτόματα ένας αριθμός σημείων.
plot_points = n	καθορίζει ότι σε κάθε άξονα θα χρησιμοποιηθούν n σημεία
plot_points = (nx, ny)	καθορίζει ότι στον άξονα x θα χρησιμοποιηθούν nx σημεία, ενώ στον άξονα y θα χρησιμοποιηθούν ny σημειά.

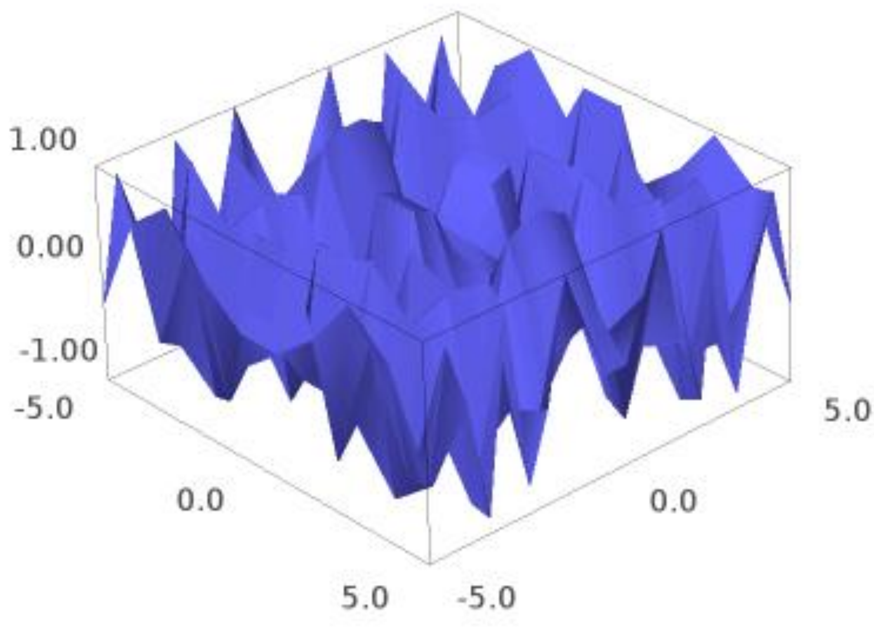
In[3]: plot3d(sin(x^2 + y^2), (x – 5, 5), (y, -5, 5), plot_points=50)

Out[3]:



In[4]: plot3d(sin(x^2 + y^2), (x – 5, 5), (y, -5, 5), plot_points=(10, 20))

Out[4]:



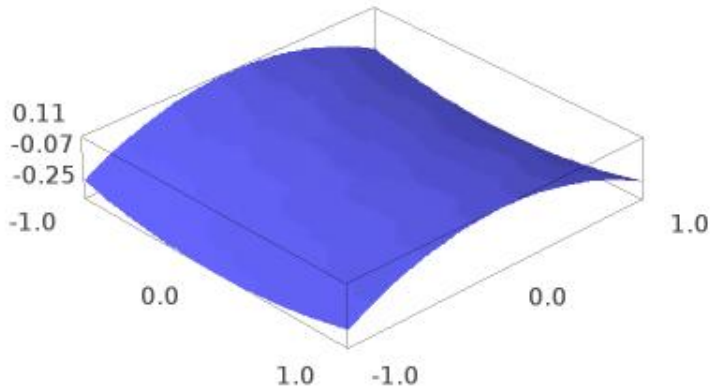
3.2.3 aspect_ratio

Η επιλογή ***aspect_ratio*** καθορίζει τους λόγους των πλευρών του παραλληλεπιπέδου, μέσα στο οποίο τοποθετείται το γράφημα.

aspect_ratio = 'automatic' (προεπιλογή)	καθορίζει τους λόγους ανάλογα με τα πραγματικά μήκη των αξόνων.
aspect_ratio = (x, y, z)	καθορίζει αυστηρώς τους λόγους των πλευρών του διάφανου κουτιού (Box) ανάλογα με τα πραγματικά μήκη των αξόνων.

In[5]: plot3d(x^2/9 - y^2/4, (x, -1, 1), (y, -1, 1), aspect_ratio=(1, 1, 1))

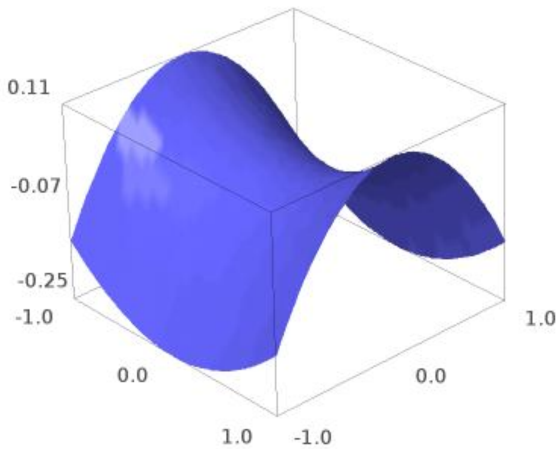
Out[5]:



Αυτό βέβαια δεν είναι πάντα επιθυμητό γιατί χάνουμε κάποια χαρακτηριστικά της επιφάνειας. Π.χ δεν φαίνεται καθαρά το «σαμαρωτό» σχήμα της επιφάνειας. Με **aspect_ratio = (1,1,4)** θα έχουμε καλύτερο αποτέλεσμα.

In[6]: plot3d(x^2/9 - y^2/4, (x, -1, 1), (y, -1, 1), aspect_ratio=(1, 1, 4))

Out[6]:



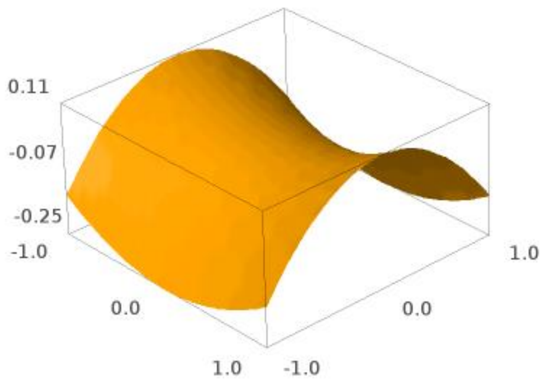
3.2.4 color

Η επιλογή **color** καθορίζει το χρώμα με το οποίο θα σχεδιασθεί η γραφική παράσταση.

color = (r, g, b)	καθορίζει το χρώμα ως συνδυασμός τριών χρωμάτων, του κόκκινου (r), του πράσινου (g) και του μπλε (b). Δίνοντας σε κάθε ένα από τα r, g, b, τιμές στο διάστημα [0, 1]. Εξ' ορισμού έχει την τιμή (0, 0, 1) που αντιστοιχεί στο μπλε χρώμα.
color = string	το χρώμα καθορίζεται με τη χρήση μίας συμβολοσειράς π.χ. 'red', 'blue', 'purple' κ.τ.λ.

In[7]: plot3d(x^2/9 - y^2/4, (x, -1, 1), (y, -1, 1), color ='orange')

Out[7]:



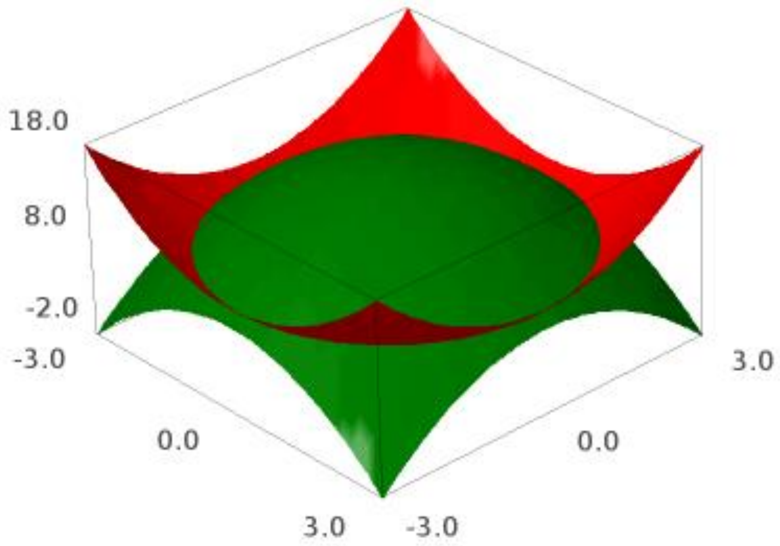
3.3. Σχεδίαση δύο ή περισσοτέρων επιφανειών στο ίδιο σύστημα αξόνων.

Με τη συνάρτηση ***plot3d***, σε αντίθεση με τη συνάρτηση ***plot***, μπορούμε να σχεδιάσουμε μόνο μία τρισδιάστατη επιφάνεια. Η συνάρτηση ***show***, την οποία την είδαμε στα δισδιάστατα γραφήματα, μας επιτρέπει να εμφανίζουμε πολλές επιφάνειες στο ίδιο σύστημα αξόνων.

Στο επόμενο παράδειγμα φαίνεται η τομή δύο παραβολοειδών.

In[8]:
```
g1=plot3d(x^2+y^2, (x, -3, 3), (y, -3, 3), color='red')
g2=plot3d(16 - (x^2+y^2),(x, -3, 3), (y, -3, 3), color='green')
show(g1+g2)
```

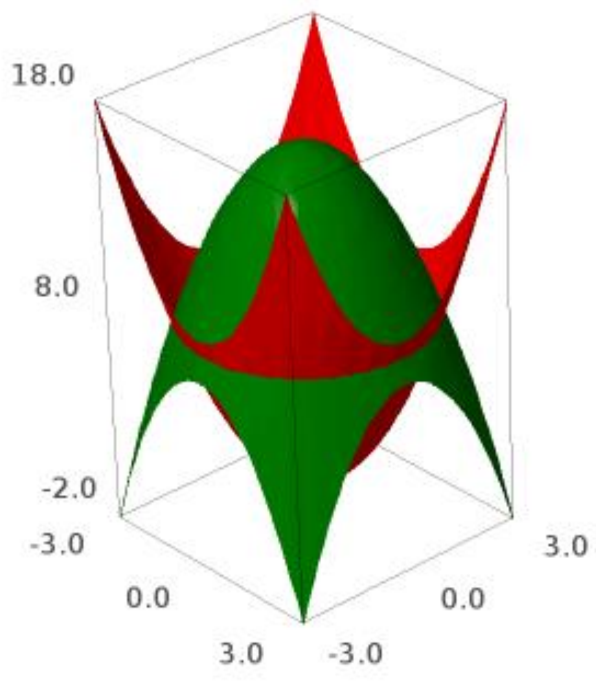
Out[8]:



Επειδή τα δύο παραβολοειδή δεν φαίνονται καθαρά μπορούμε με την επιλογή aspect_ratio να αλλάξουμε την αναλογία των αξόνων έτσι ώστε να είναι ξεκάθαρο ότι σχεδιάσαμε δύο παραβολοειδή.

In[9]:
```
g1=plot3d(x^2+y^2, (x, -3, 3), (y, -3, 3), color='red')
g2=plot3d(16 - (x^2+y^2),(x, -3, 3), (y, -3, 3), color='green')
show(g1+g2, aspect_ratio=(1, 1, 0.5))
```

Out[9]:



3.4 Επιπλέον συναρτήσεις για τρισδιάστατα γραφικά

3.4.1 Γραφική παράσταση επιφάνειας που ορίζεται παραμετρικά

Τη γραφική παράσταση μιας επιφάνειας, που ορίζεται με την βοήθεια παραμέτρων την παίρνουμε με την εντολή ***parametric_plot3d***. Τέτοιες επιφάνειες λέγονται **παραμετρικές** επιφάνειες. Αυτές οι επιφάνειες μπορεί να τέμνουν τον εαυτό τους ή να κουλουριάζουν γύρω από κάποιο άξονα κ.ο.κ. Τέτοιες δυνατότητες δεν έχει η ***plot3d***.

parametric_plot3d((fx, fy, fz), (u, umin, umax))

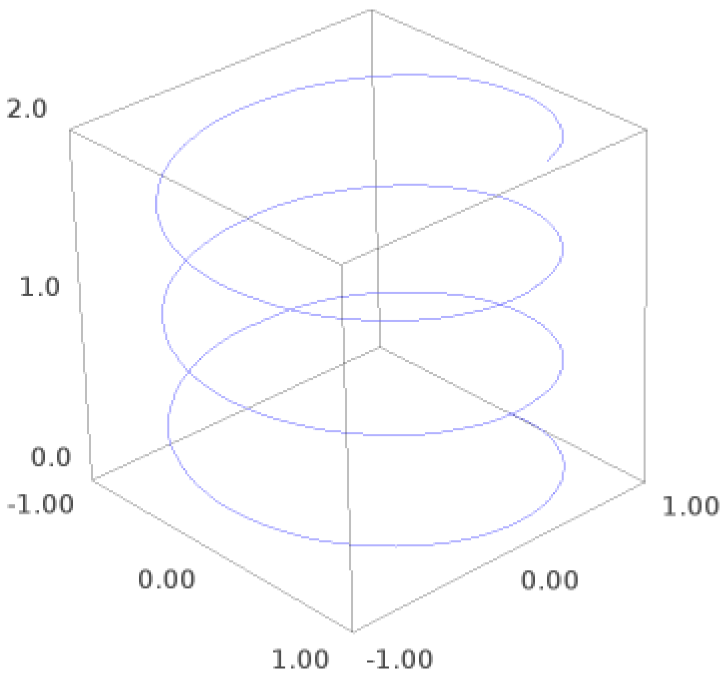
όπου fx, fy, fz συναρτήσεις της μεταβλητής u. Σχεδιάζει τη γραφική παράσταση της συνάρτησης που ορίζεται παραμετρικά ως προς την παραμέτρου u πάνω στο διάστημα [umin, umax].

parametric_plot3d((fx, fy, fz), (u, umin, umax), (v, vmin, vmax))

όπου fx, fy, fz συναρτήσεις των μεταβλητών u και v. Σχεδιάζει τη γραφική παράσταση της συνάρτησης που ορίζεται παραμετρικά ως προς τις παραμέτρου u, v πάνω στο ορθογώνιο που ορίζουν τα διαστήματα [umin, umax] και [vmin, vmax].

In[10]: u = var('u')
parametric_plot3d((sin(u), cos(u), u/10), (u, 0, 20))

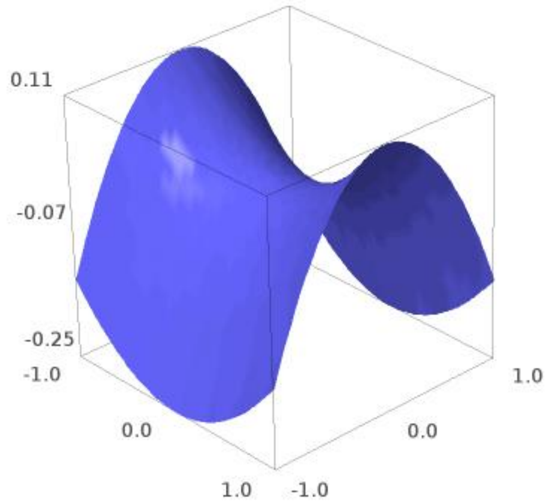
Out[10]:



Π.χ αν θέλουμε να σχεδιάσουμε το σαμάρι δεν έχουμε παρά να θέσουμε τον x ίσο με την πρώτη παράμετρο u, το y ίσο με την δεύτερη παράμετρο v και το z (δηλ. το ύψος f[x,y]) ίσο με f[u,v]:

In[11]: v = var('v')
parametric_plot3d((u, v, u^2/9-v^2/4), (u, -1, 1), (v, -1, 1))

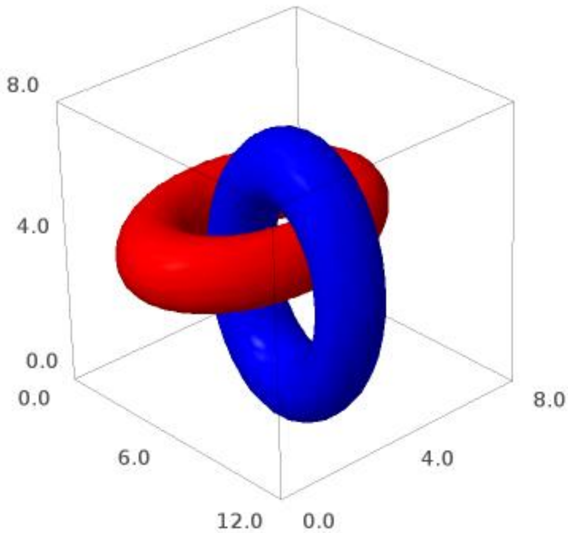
Out[11]:



Με παραμετρικές εξισώσεις μπορούμε να ορίσουμε (και να σχεδιάσουμε κατά συνέπεια) τον τόρο, διάφορες κυλινδρικές επιφάνειες, παραβολοειδή, την σφαίρα κ.ο.κ.

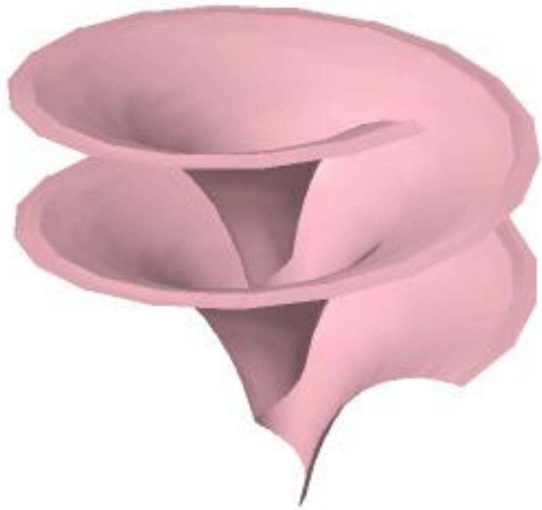
In[12]: f1 = (4+(3+cos(v))*sin(u), 4+(3+cos(v))*cos(u), 4+sin(v))
f2 = (8+(3+cos(v))*cos(u), 3+sin(v), 4+(3+cos(v))*sin(u))
p1 = parametric_plot3d(f1, (u,0,2*pi), (v,0,2*pi), color="red")
p2 = parametric_plot3d(f2, (u,0,2*pi), (v,0,2*pi), color="blue")
p1 + p2

Out[12]:



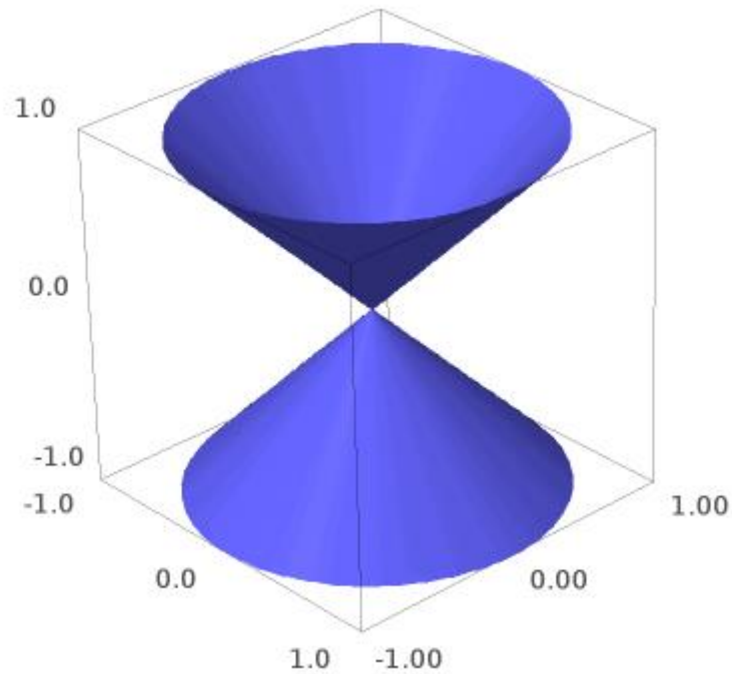
In[13]:
```
fx = cos(u)*sin(v)
fy = sin(u)*sin(v)
fz = (cos(v)+log(tan(v/2))) + 0.2*u
parametric_plot3d((fx, fy, fz), (u, 0, 12.4), (v, 0.1, 2),frame=False,
                                                     color='pink')
```

Out[13]:



In[14]:
```
parametric_plot3d((u*cos(v), u*sin(v), u), (u, -1, 1), (v, 0, 2*pi+0.5),
                                                   plot_points=[50,50])
```

Out[14]:



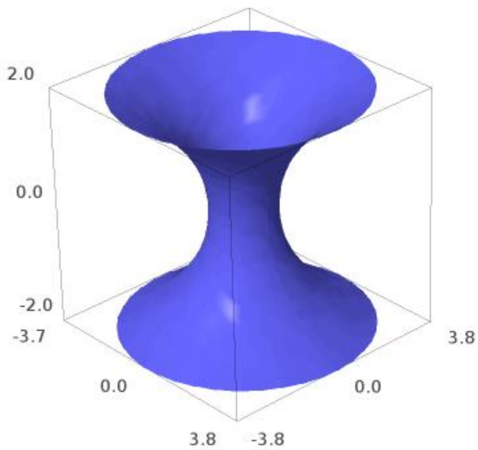
3.4.2 Γραφική παράσταση επιφάνειας που ορίζεται με κυλινδρικές συντεταγμένες

Τη γραφική παράσταση μιας επιφάνειας, που ορίζεται με την βοήθεια κυλινδρικών συντεταγμένων την παίρνουμε με την εντολή ***cylindrical_plot3d***.

cylindrical_plot3d(f, (u, umin, umax), (v, vmin, vmax))	
	όπου f συνάρτηση δύο μεταβλητών u και v που αντιστοιχούν στις κυλινδρικές συντεταγμένες. Σχεδιάζει τη γραφική παράσταση της συνάρτησης πάνω στο ορθογώνιο που ορίζουν τα διαστήματα [umin, umax] και [vmin, vmax].

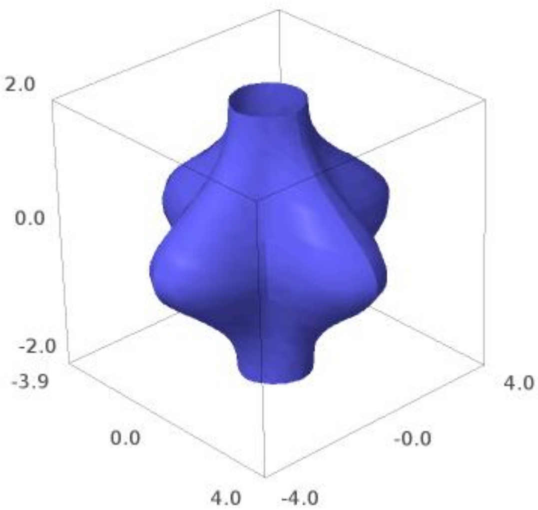
In[15]: theta, z = var('theta, z')
cylindrical_plot3d(cosh(z), (theta, 0, 2*pi), (z, -2, 2))

Out[15]:



In[16]: cylindrical_plot3d(e^(-z^2)*(cos(4*theta)+2)+1, (theta, 0, 2*pi), (z, -2, 2))

Out[16]:



3.4.3 Γραφική παράσταση επιφάνειας που ορίζεται με σφαιρικές συντεταγμένες

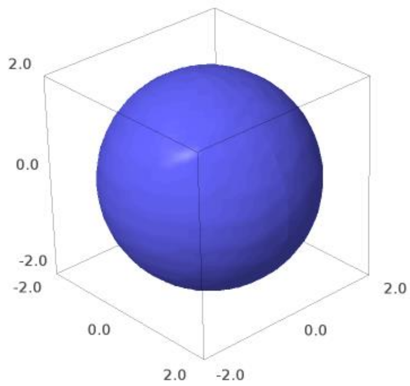
Τη γραφική παράσταση μιας επιφάνειας, που ορίζεται με την βοήθεια σφαιρικών συντεταγμένων την παίρνουμε με την εντολή ***spherical_plot3d***.

spherical_plot3d(f, (u, umin, umax), (v, vmin, vmax))	
	όπου f συνάρτηση δύο μεταβλητών u και v που αντιστοιχούν στις σφαιρικές συντεταγμένες. Σχεδιάζει τη γραφική παράσταση της συνάρτησης πάνω στο ορθογώνιο που ορίζουν τα διαστήματα [umin, umax] και [vmin, vmax].

Αρχικά ας σχεδιάσουμε μία σφαίρα ακτίνας 2.

In[17]: phi = var('phi')
spherical_plot3d(2, (theta, 0, 2*pi), (phi, 0, pi))

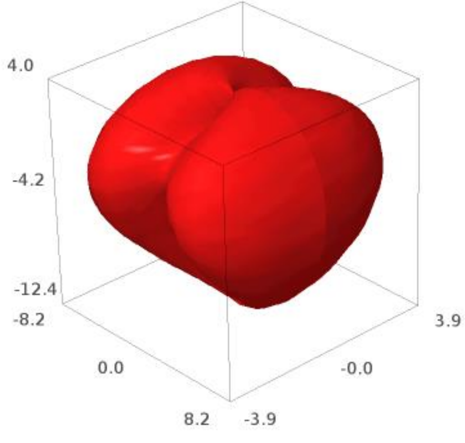
Out[17]:



Στη συνέχεια θα σχεδιάσουμε κάτι που μοιάζει με καρδιά!!!

In[18]: spherical_plot3d((2+cos(2*theta)) * (phi+1), (theta, 0, 2*pi), (phi, 0, pi), color=(1, 0.1, 0.1))

Out[18]:



3.4.4 Γραφική παράσταση επιφάνειας που ορίζεται πεπλεγμένα

Τη γραφική παράσταση μιας επιφάνειας, που ορίζεται πεπλεγμένα την παίρνουμε με την εντολή ***implicit_plot3d***.

implicit_plot3d(f, (x, xmin, xmax), (y, ymin, ymax), (z, zmin, zmax))

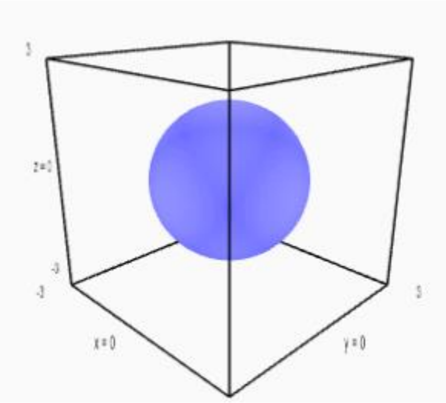
Σχεδιάζει τη γραφική παράσταση της f για τα διαστήματα (xxmin, xmax), (ymin, ymax) και (zmin, zmax). Όπου f μία συνάρτηση ή μία ισότητα τριών μεταβλητών x, y και z. Αν η f είναι συνάρτηση, σχεδιάζει τη γραφική παράσταση της f(x, y,z) = 0.

Προσοχή: Στον ορισμό της ισότητας το ίσον γράφεται ως == (με δύο ίσον).

Αρχικά ας σχεδιάσουμε μία σφαίρα ακτίνας 2.

In[19]: z=var('z')
implicit_plot3d(x^2+y^2+z^2==4, (x, -3, 3), (y, -3,3), (z, -3,3))

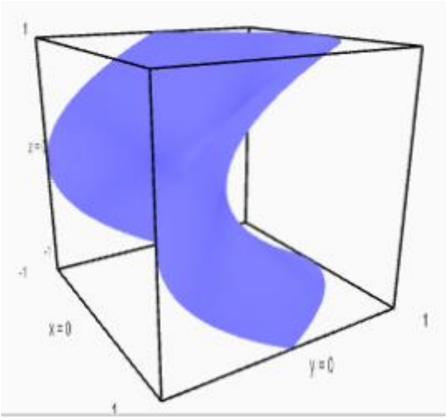
Out[19]:



Στη συνέχεια θα σχεδιάσουμε μία απλή υπερβολική επιφάνεια.

In[20]: implicit_plot3d(x*x + y - z*z, (x, -1, 1), (y, -1, 1), (z, -1, 1))

Out[20]:



3.4.5 Τρισδιάστατα γραφικά στο επίπεδο

Ορίζουμε ως ισοϋψή καμπύλη μιας συνάρτησης δύο μεταβλητών f(x, y) το δισδιάστατο γράφημα της εξίσωσης f(x, y) = k για κάποια τιμή του k. Το υψομετρικό γράφημα (contour plot) είναι ένα σύνολο ισοϋψών καμπυλών που σχεδιάζονται στο ίδιο σύστημα αξόνων.

Η συνάρτηση ***contour_plot*** του Sage σχεδιάζει υψομετρικά γραφήματα συναρτήσεων δύο μεταβλητών. Οι ισοϋψείς καμπύλες ενώνουν σημεία της επιφάνειας του υψομετρικού γραφήματος που έχουν το ίδιο ύψος.

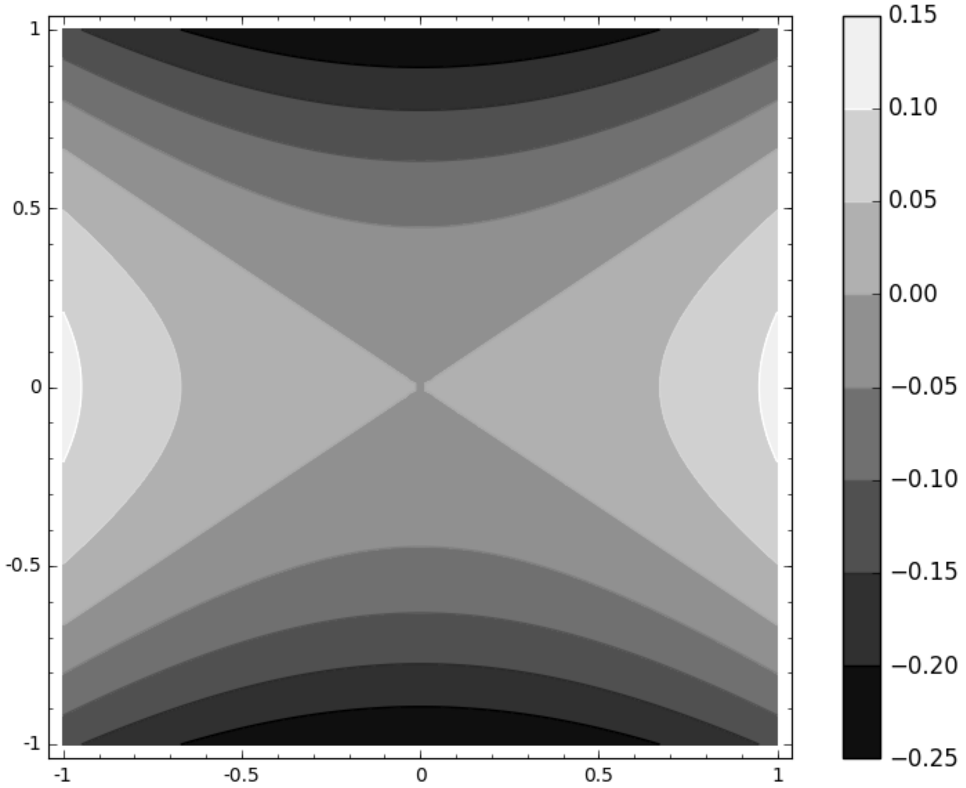
contour_plot(f(x, y), (x, xmin, xmax), (ymin,ymax))	
	Σχεδιάζει το υψομετρικό γράφημα της συνάρτησης f(x,y) μέσα σε ένα ορθογώνιο πλαίσιο που ορίζεται από τις συντεταγμένες xmin, xmax, και ymin και ymax.

Η σχεδίαση των υψομετρικών γραφημάτων στο Sage γίνεται με σκίαση, με τέτοιο τρόπο ώστε οι περιοχές που αντιστοιχούν σε μεγαλύτερες τιμές της f(x, y) (δηλαδή σε μεγαλύτερα ύψη) να εμφανίζονται πιο φωτεινές. Όπως και με όλες τις συναρτήσεις γραφικών του Sage, έτσι και με τη συγκεκριμένη μπορούμε να χρησιμοποιήσουμε και ένα σύνολο επιλογών για να ελέγχουμε την εμφάνιση των γραφημάτων. Ας δούμε κάποιες που είναι διαφορετικές από τις επιλογές των άλλων συναρτήσεων σχεδίασης γραφημάτων:

contours = n	καθορίζει τον αριθμό των ισοϋψών καμπυλών που θα σχεδιαστούν. Η προεπιλογή είναι 8 ισοκατανεμημένες ισοϋψής καμπύλες.
contours = (k1, k2, …)	σχεδιάζει τις ισοϋψείς καμπύλες που αντιστοιχούν στις τιμές k1, k2, …
colorbar = True	εμφανίζει δίπλα στο γράφημα μία μπάρα στην οποία αντιστοιχεί τις διάφορες αποχρώσεις τιμές.
colorbar = False (προεπιλογή)	δεν εμφανίζει την μπάρα των αποχρώσεων.

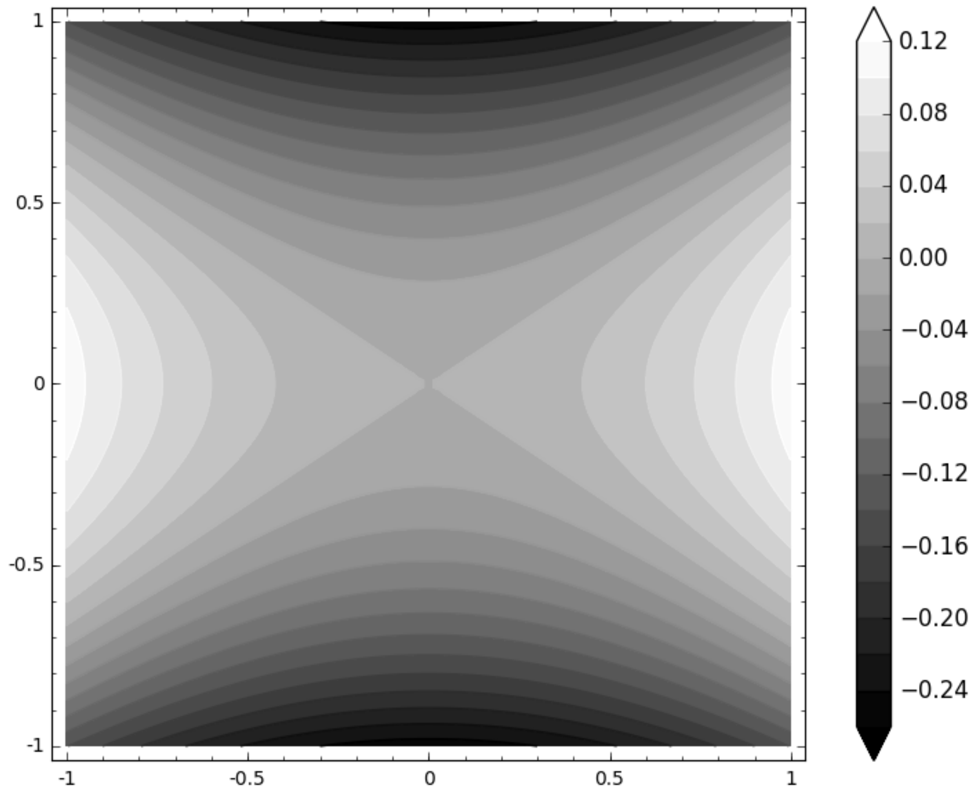
In[21]: contour_plot(x^2/9-y^2/4 , (x, -1, 1), (y, -1, 1), colorbar=True)

Out[21]:



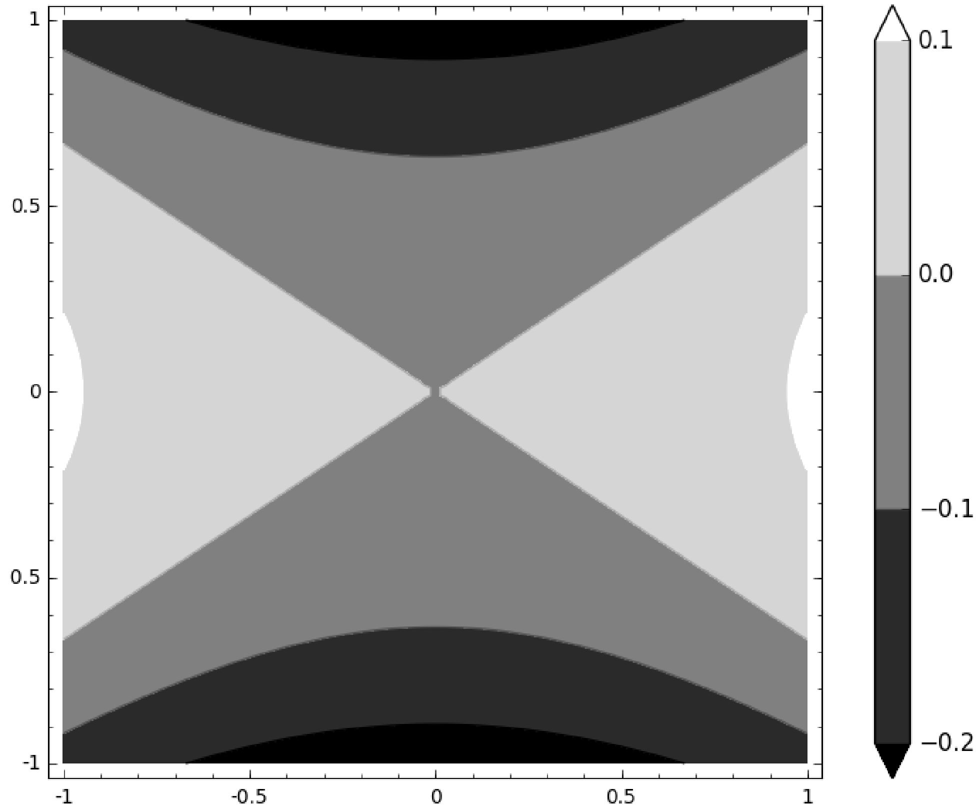
In[22]: contour_plot(x^2/9-y^2/4 , (x, -1, 1), (y, -1, 1), colorbar=True, contours=20)

Out[22]:



In[23]: contour_plot(x^2/9-y^2/4 , (x, -1, 1), (y, -1, 1), colorbar=True, contours=(–0.2, –0.1, 0, 0.1))

Out[23]:



Χρήστος Τσαγγάρης
ΕΔΙΠ Τμήματος Μαθηματικών

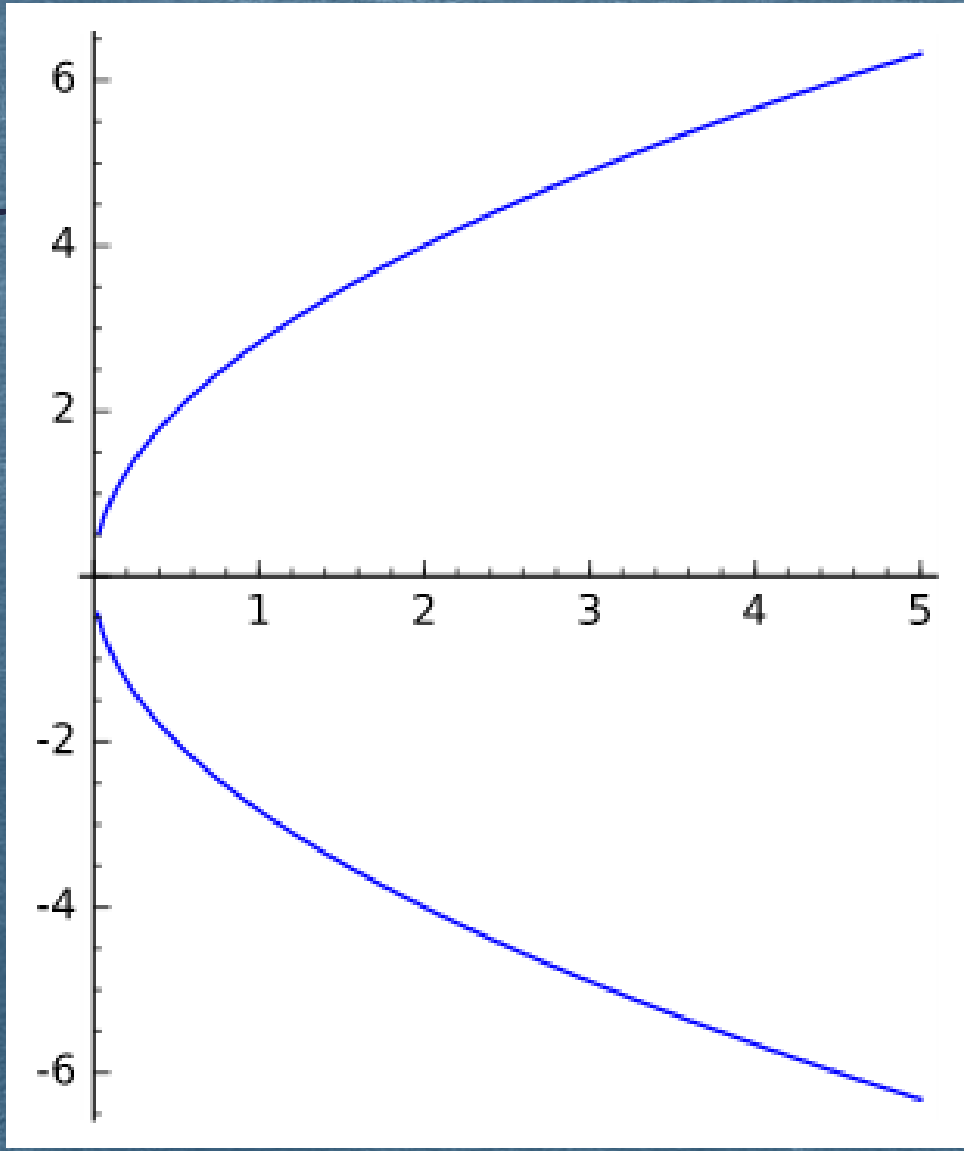
23-11-2016

Παραβολή

Άσκηση 1η: Να σχεδιάσετε την παραβολή $y^2=8x$.

Λύση 1η:

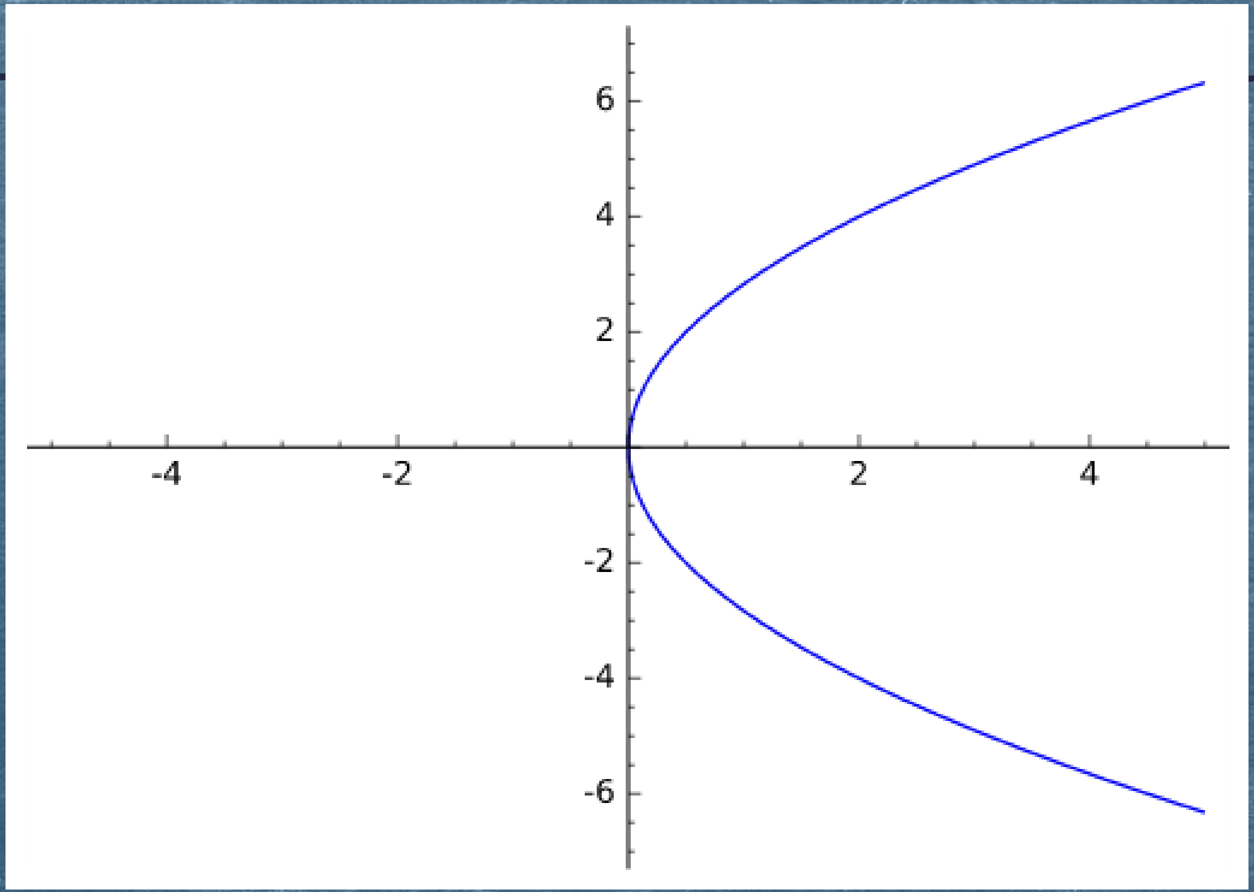
```
plot((sqrt(8*x), -sqrt(8*x)), (x, -5, 5), aspect_ratio=0.5)
```


6
4
2
-2
-4
-6
1
2
3
4
5

Άσκηση 1η: Συνέχεια …

Λύση 2η:

```
y=var('y')

implicit_plot(y^2==8*x, (x,-5,5), (y, -7, 7), axes=True, frame=False,
                                                          aspect_ratio=0.5)
```

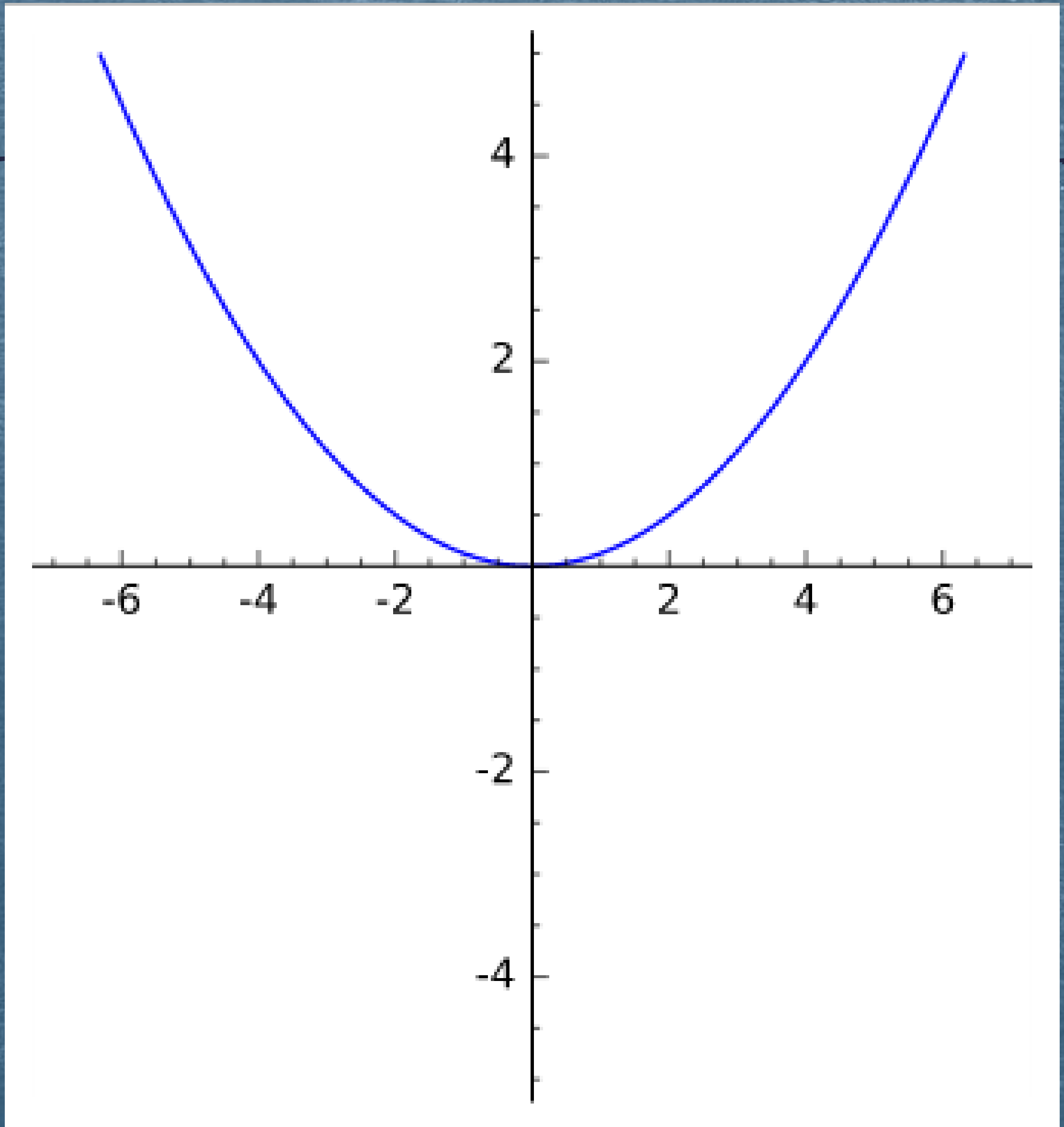


6
4
2
-2
-4
-6
-4
-2
2
4

Άσκηση 2η: Να σχεδιάσετε την παραβολή $x^2=8y$.

Λύση:

```
implicit_plot(x^2==8*y, (x,-7,7), (y, -5, 5), axes=True, frame=False,
                                                aspect_ratio=1.5)
```

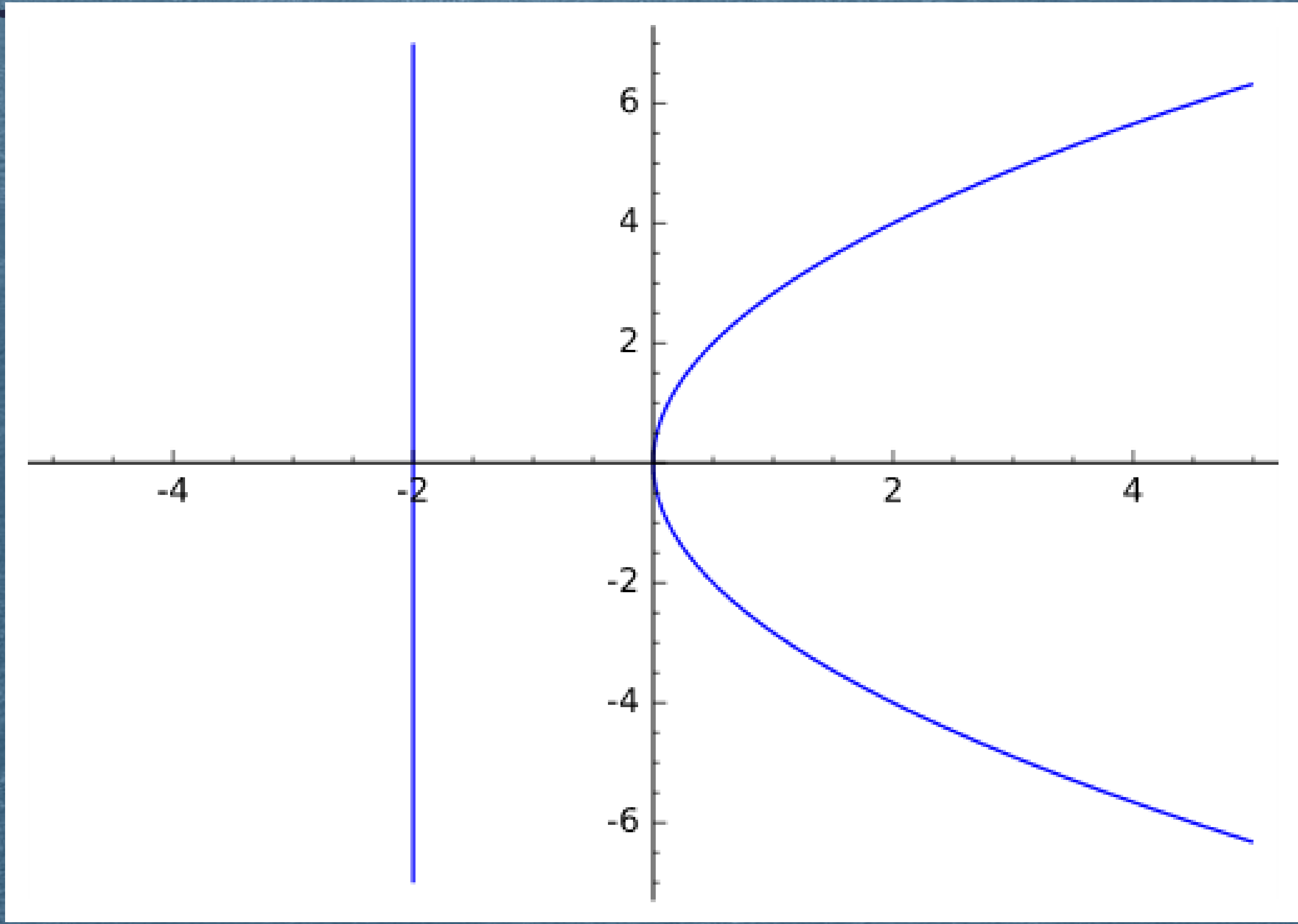


4
2
-2
-4
-6
-4
-2
2
4
6

Άσκηση 3η: Να σχεδιάσετε την παραβολή $y^2=8x$ μαζί με τη διευθετούσα της.

Λύση:

```
g1=implicit_plot(y^2==8*x, (x,-5,5), (y, -7, 7))

g2=implicit_plot(x==-2, (x,-5,5), (y, -7, 7))

show(g1+g2, axes=True, frame=False, aspect_ratio=0.5)
```

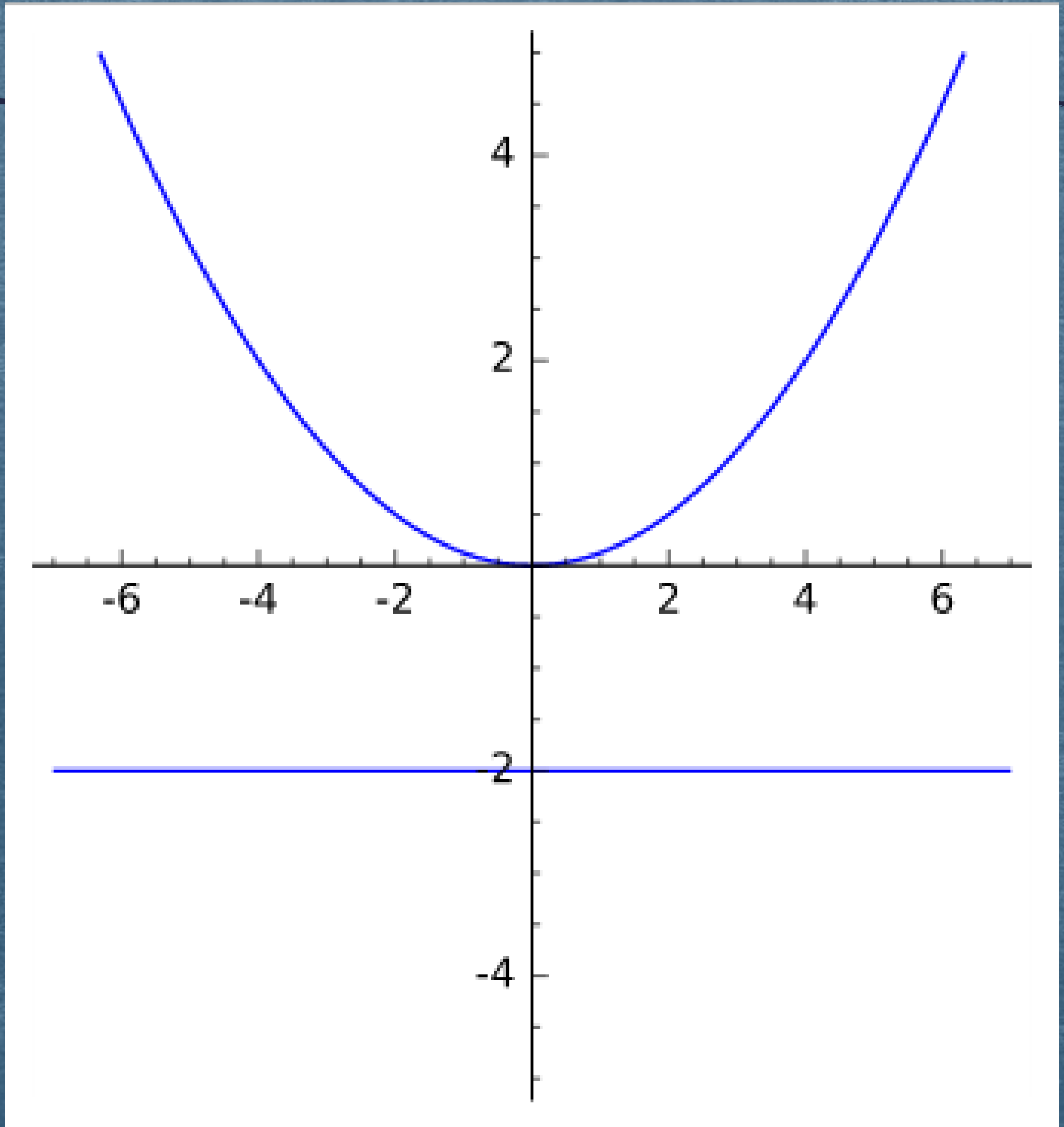


6
4
2
-4
-2
2
4
-2
-4
-6

Άσκηση 4η: Να σχεδιάσετε την παραβολή $x^2=8y$ μαζί με τη διευθετούσα της.

Λύση:

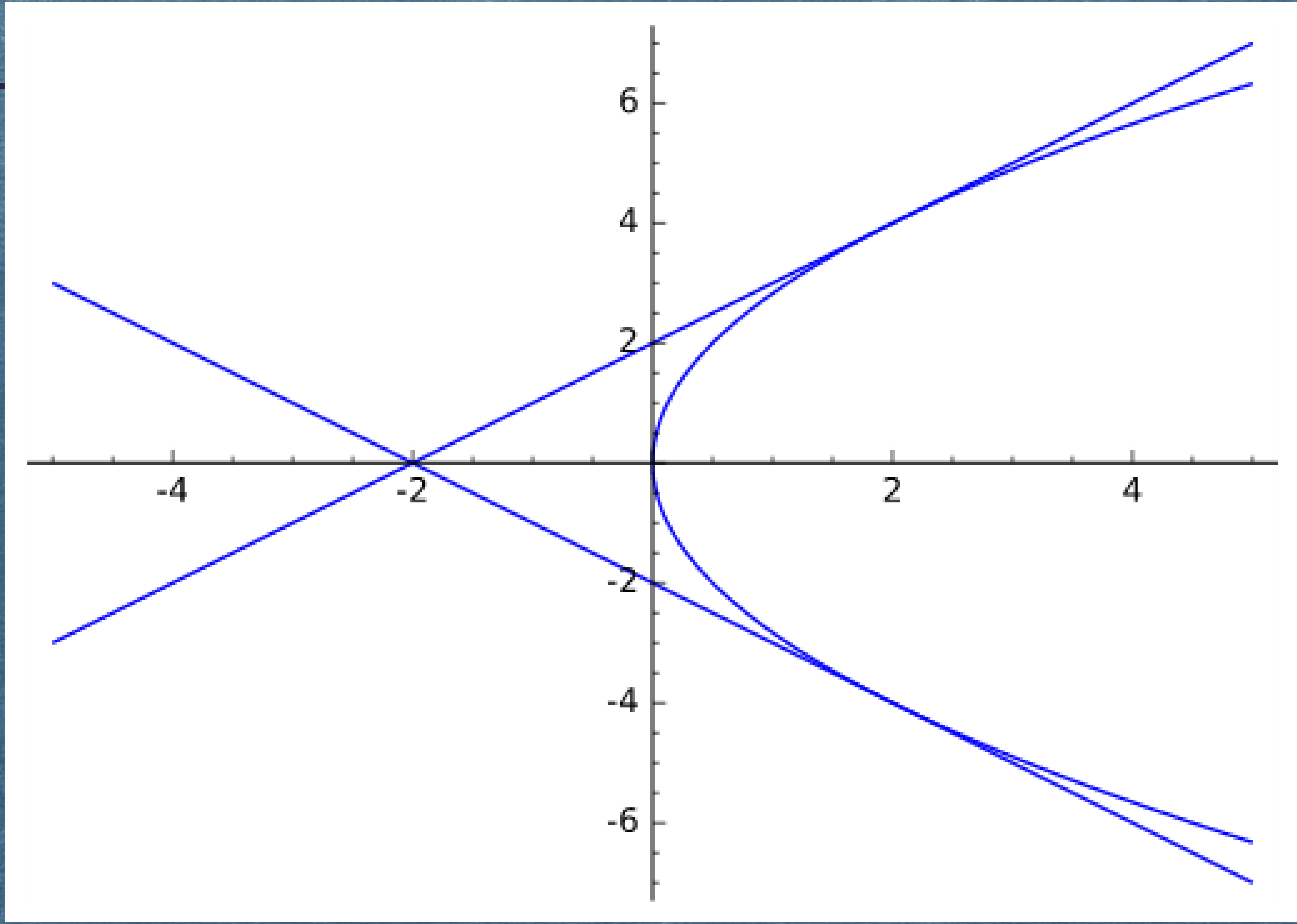
```
g1=implicit_plot(x^2==8*y, (x,-7,7), (y, -5, 5))

g2=implicit_plot(y==-2, (x,-7,7), (y, -5, 5))

show(g1+g2, axes=True, frame=False, aspect_ratio=1.5)
```


4
2
-2
-4
-6
-4
-2
2
4
6

Άσκηση 5η: Να σχεδιάσετε την παραβολή $y^2=8x$ καθώς και την εφαπτομένη της στο σημείο (2, 4) και στο σημείο (2, -4).

Λύση:

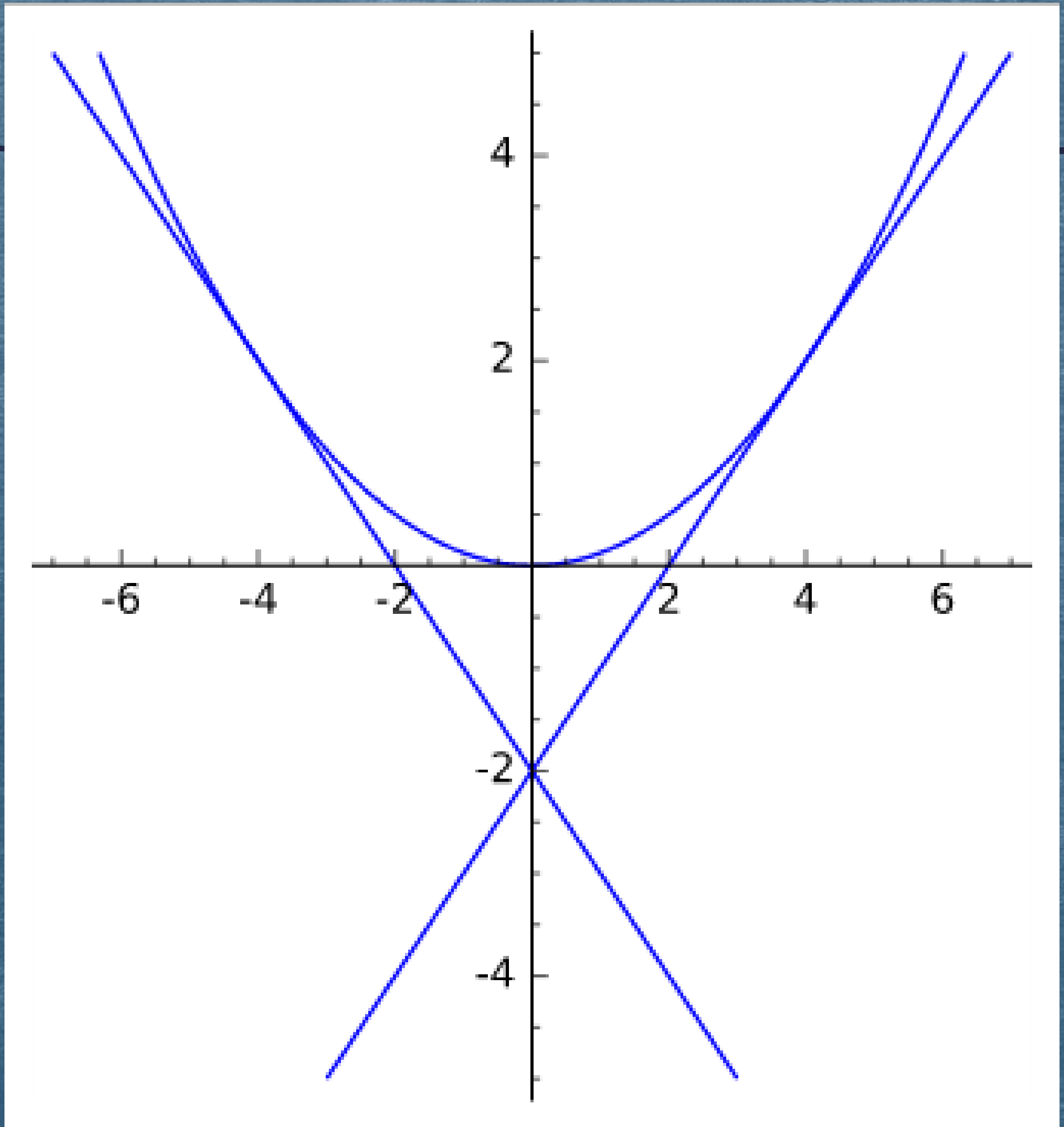
```
g1=implicit_plot(y^2==8*x, (x,-5,5), (y, -7, 7))

g2=implicit_plot(y*4==4*(x+2), (x,-5,5), (y, -7, 7))

g3=implicit_plot(y*(-4)==4*(x+2), (x,-5,5), (y, -7, 7))

show(g1+g2+g3, axes=True, frame=False, aspect_ratio=0.5)
```


6
4
2
-2
-4
-6
-4
-2
2
4

Άσκηση 6η: Να σχεδιάσετε την παραβολή $x^2=8y$ καθώς και την εφαπτομένη της στο σημείο (4, 2) και στο σημείο (4, -2).

Λύση:

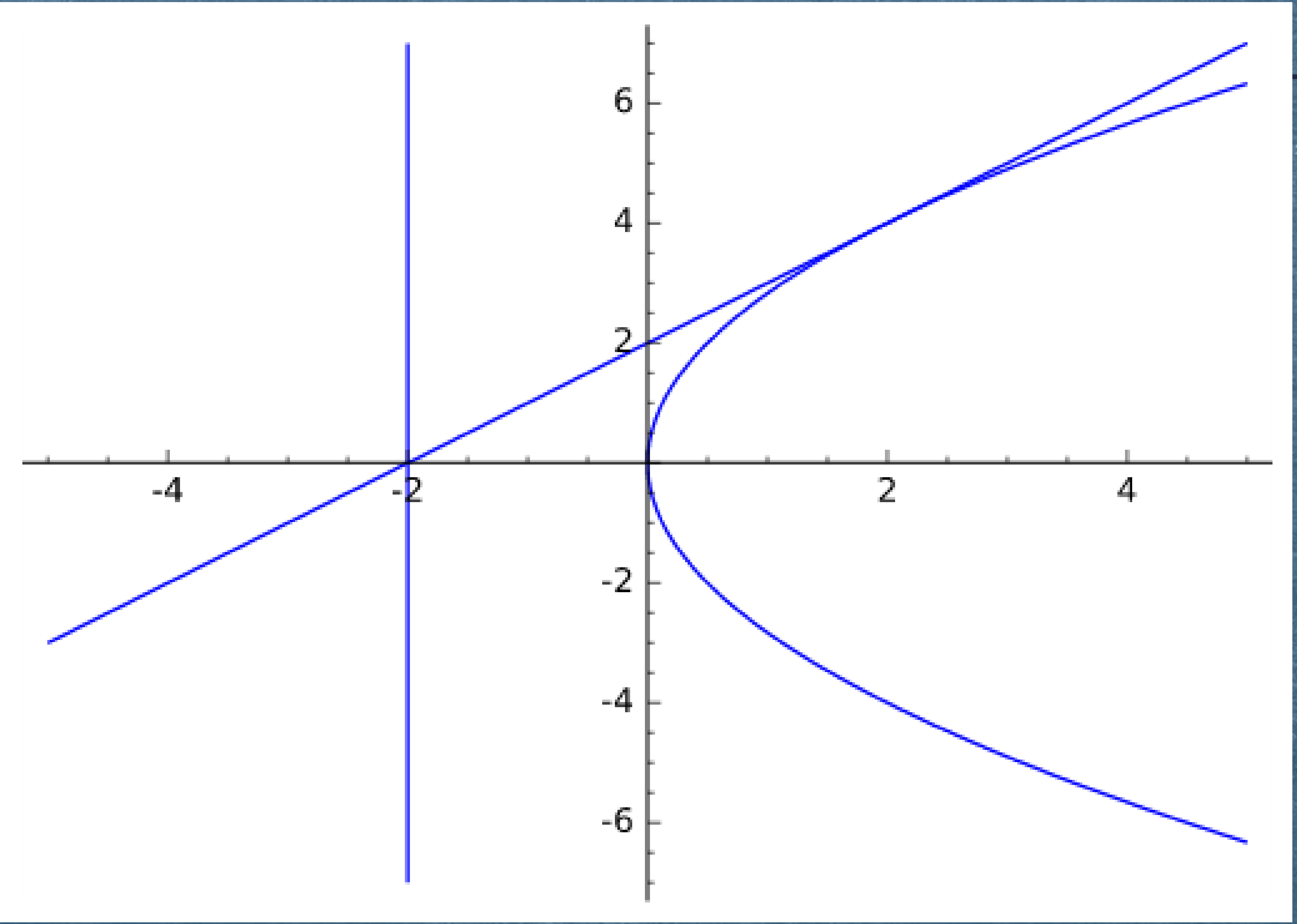
```
g1=implicit_plot(x^2==8*y, (x,-7,7), (y, -5, 5))

g2=implicit_plot(x*4==4*(y+2), (x,-7,7), (y, -5, 5))

g3=implicit_plot(x*(-4)==4*(y+2), (x,-7,7), (y, -5, 5))

show(g1+g2+g3, axes=True, frame=False, aspect_ratio=1.5)
```


4
2
-2
-4
-6
-4
-2
2
4
6

Άσκηση 7η: Να σχεδιάσετε την παραβολή $y^2=8x$ μαζί με τη διευθετούσα της, καθώς και την εφαπτομένη της στο σημείο (2, 4).

Λύση:

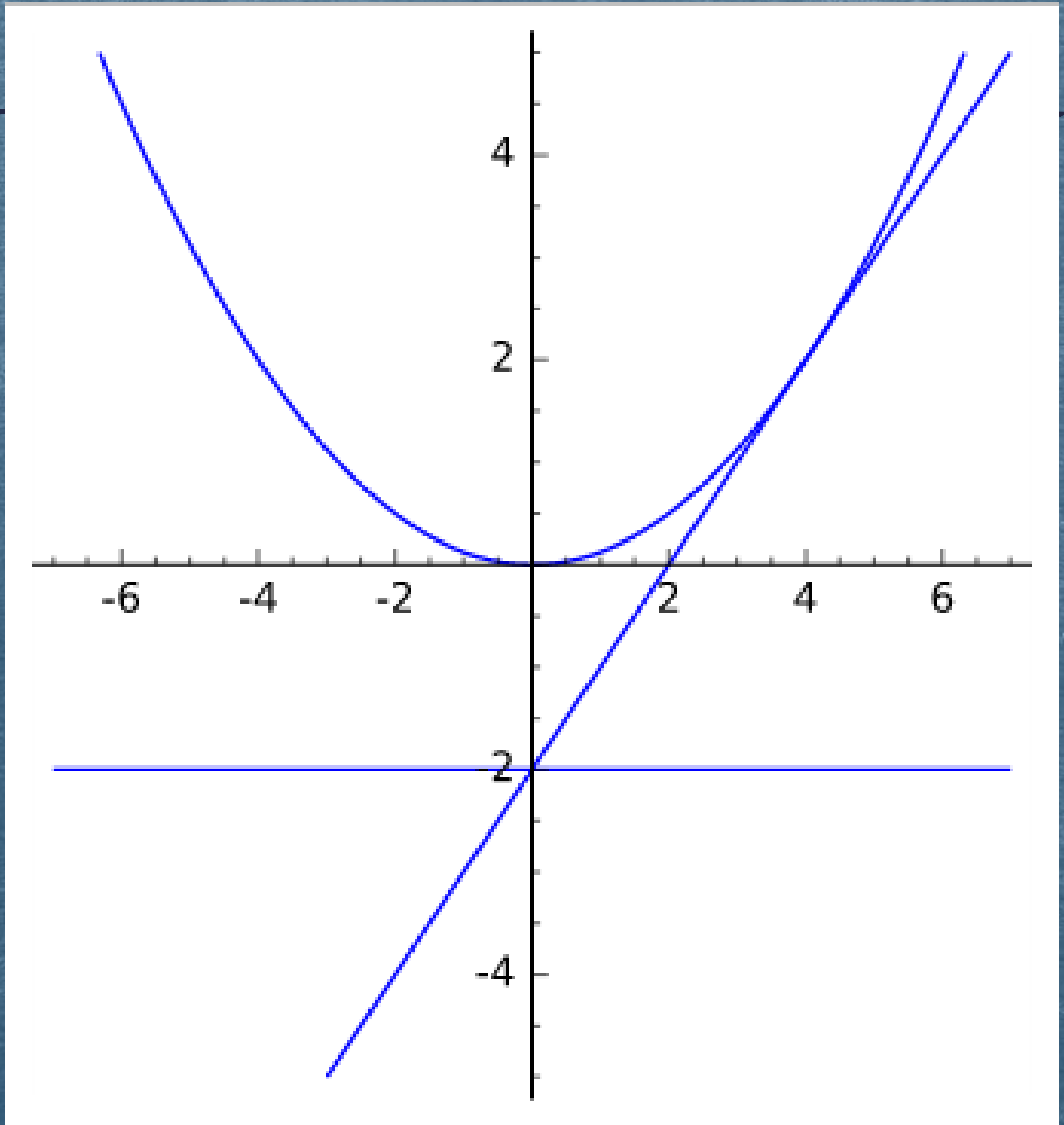
```
g1=implicit_plot(y^2==8*x, (x,-5,5), (y, -7, 7))

g2=implicit_plot(y*4==4*(x+2), (x,-5,5), (y, -7, 7))

g3=implicit_plot(x==-2, (x,-5,5), (y, -7, 7))

show(g1+g2+g3, axes=True, frame=False, aspect_ratio=0.5)
```


6
4
2
-2
-4
-6
-4
-2
2
4

Άσκηση 8η: Να σχεδιάσετε την παραβολή $x^2=8y$ μαζί με τη διευθετούσα της, καθώς και την εφαπτομένη της στο σημείο (4, 2).

Λύση:

```
g1=implicit_plot(x^2==8*y, (x,-7,7), (y, -5, 5))

g2=implicit_plot(x*4==4*(y+2), (x,-7,7), (y, -5, 5))

g3=implicit_plot(y==-2, (x,-7,7), (y, -5, 5))

show(g1+g2+g3, axes=True, frame=False, aspect_ratio=1.5)
```



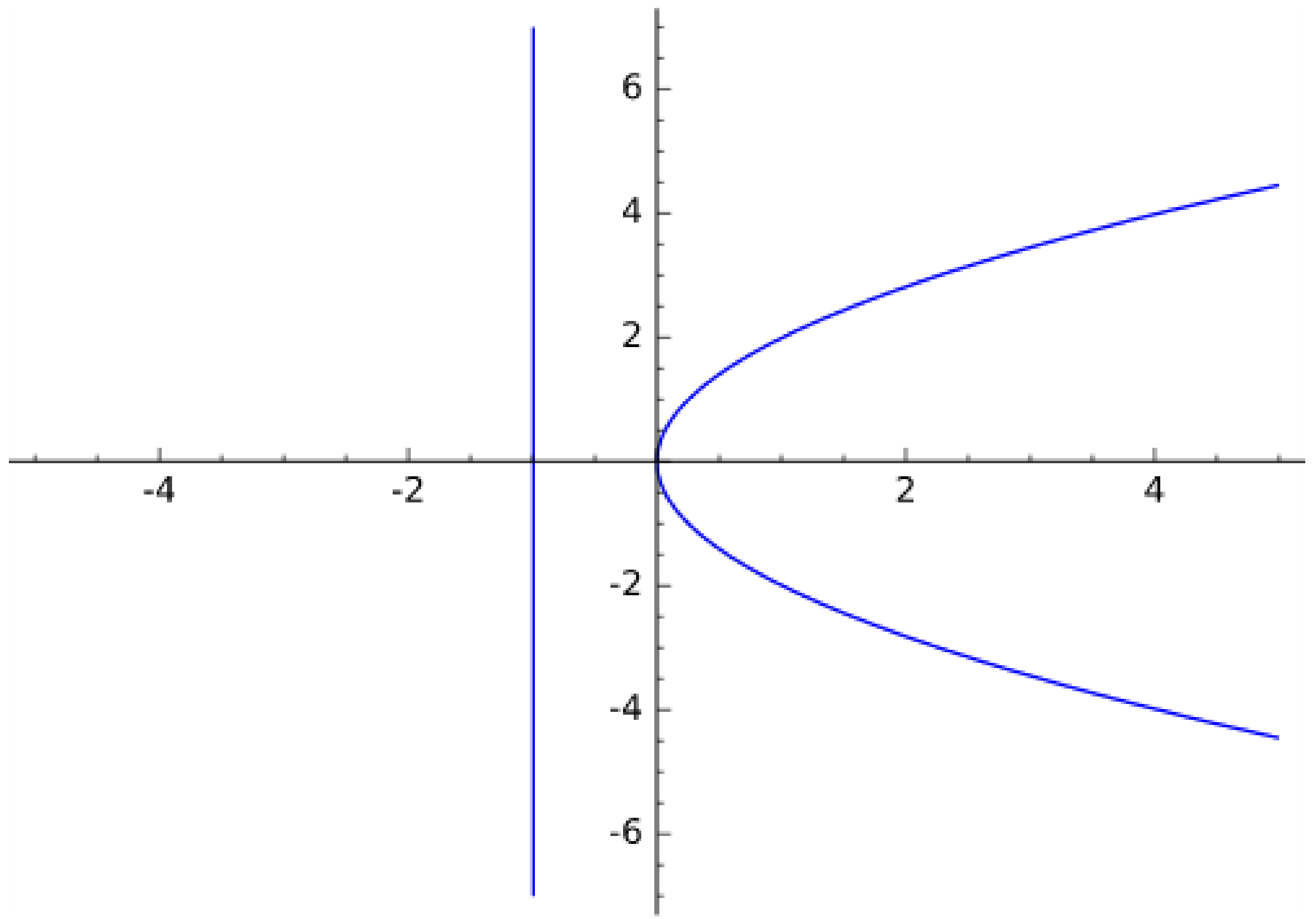
4
2
-6
-4
-2
2
4
6
-2
-4

Άσκηση 9η: Να σχεδιάσετε την παραβολή $y^2=4cx$ μαζί με τη διευθετούσα της για διάφορες τιμές της παραμέτρου c στο διάστημα [-4, 4].

Λύση:

```
@interact
def my_plot(c=(1, (-4, 4))):
    g1=implicit_plot(y^2==4*c*x, (x,-5,5), (y, -7, 7))
    g2=implicit_plot(x==-c, (x,-7,7), (y, -5, 5))
    show(g1+g2, axes=True, frame=False, aspect_ratio=0.5)
```

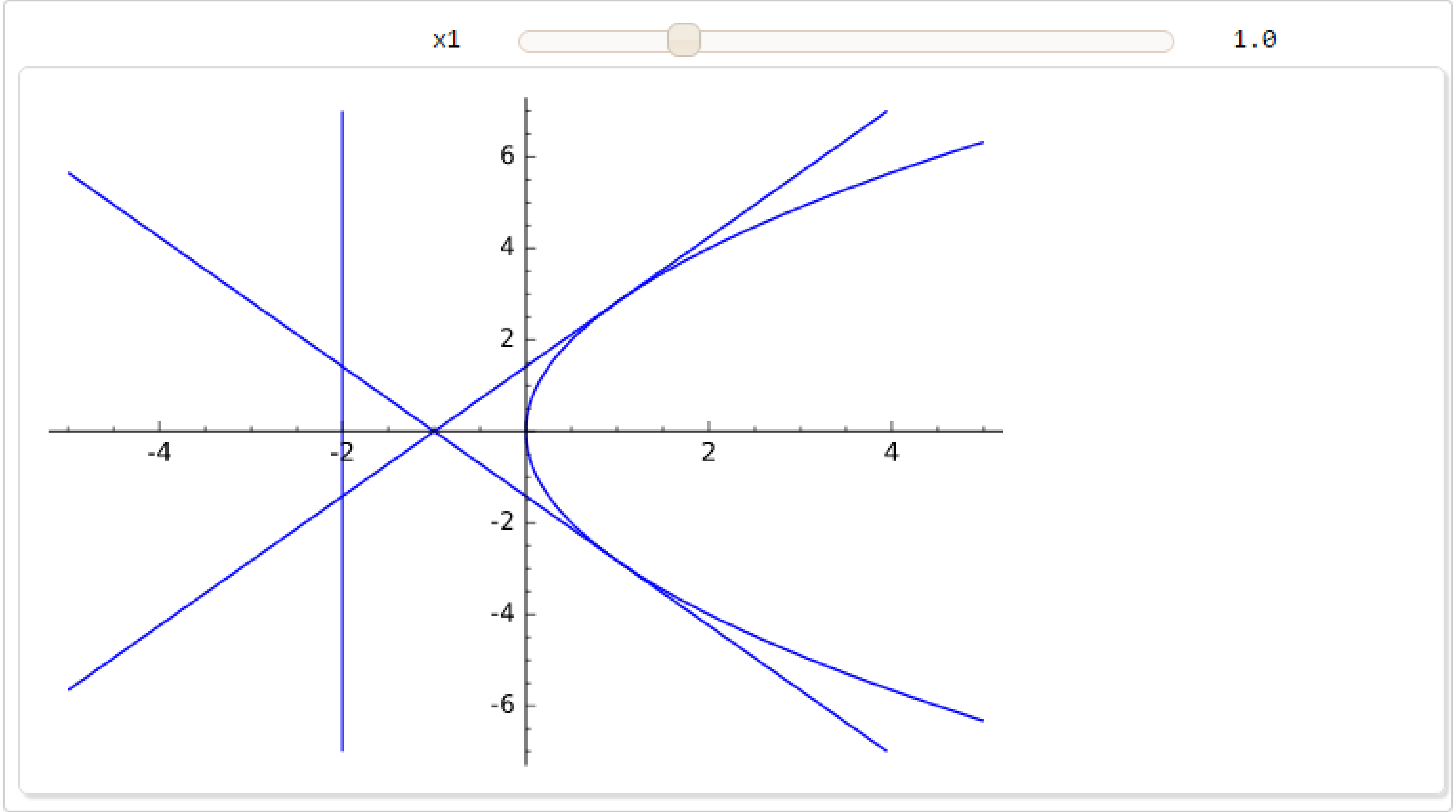
c 0.992



Άσκηση 10η: Να σχεδιάσετε την παραβολή $y^2=8x$ μαζί με τη διευθετούσα της καθώς και τις εφαπτομένες της στα συμμετρικά της σημεία.

Λύση:

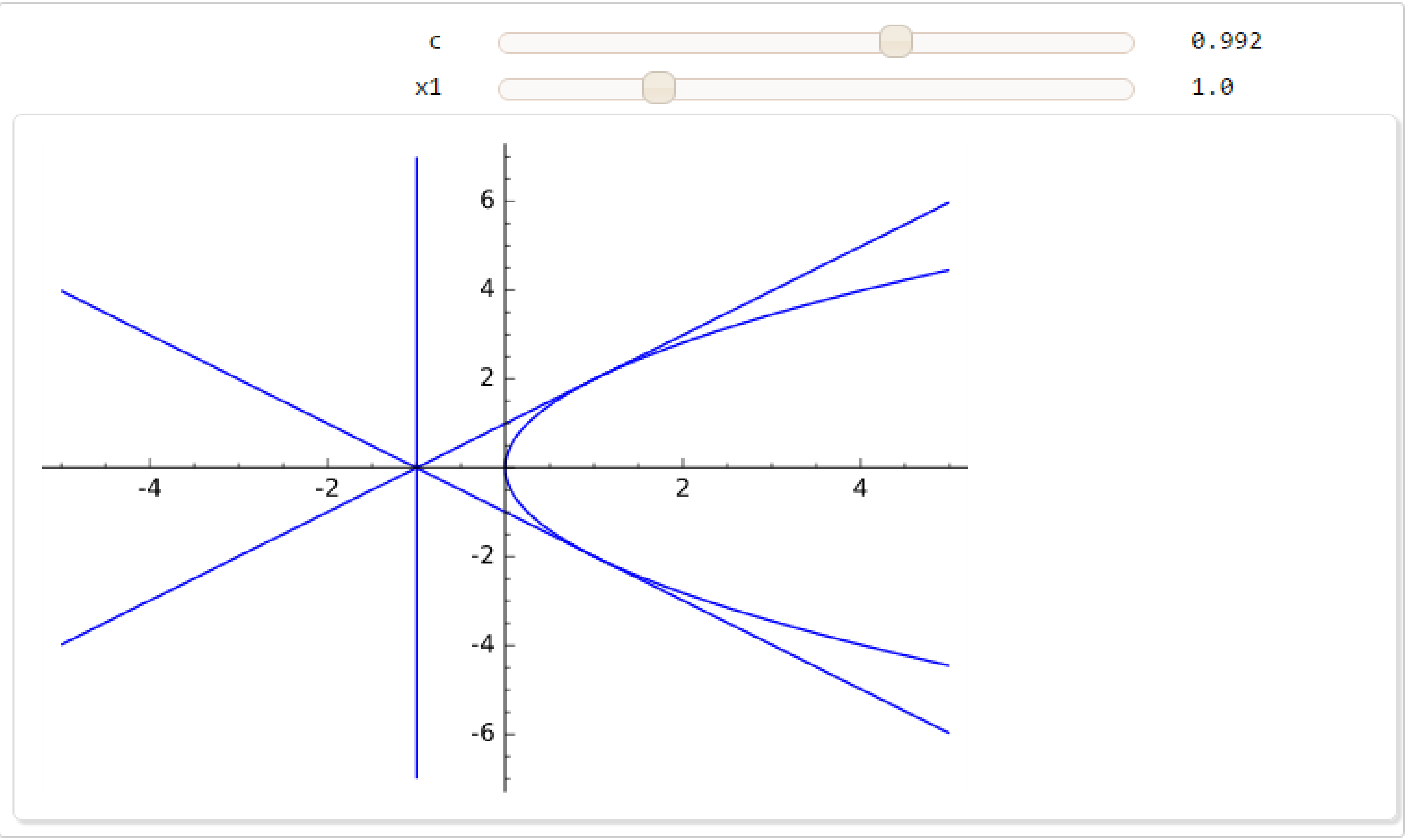
```
@interact
def my_plot(x1=(1, (0, 4))):
    g1=implicit_plot(y^2==8*x, (x,-5,5), (y, -7, 7))
    g2=implicit_plot(x==-2, (x,-5,5), (y, -7, 7))
    g3=implicit_plot(y*sqrt(8*x1)==4*(x+x1), (x,-5,5), (y, -7, 7))
    g4=implicit_plot(y*(-sqrt(8*x1))==4*(x+x1), (x,-5,5), (y, -7, 7))
    show(g1+g2+g3+g4, axes=True, frame=False, aspect_ratio=0.5)
```


x1
1.0
6
4
2
-2
-4
-6
-4
-2
2
4

Άσκηση 11η: Να σχεδιάσετε την παραβολή $y^2=4*c*x$ μαζί με τη διευθετούσα της καθώς και τις εφαπτομένες της στα συμμετρικά της σημεία, για διάφορες τιμές της παραμέτρου c στο διάστημα [-4, 4].

Λύση:

```
@interact

def my_plot(c=(1, (-4, 4)), x1=(1, (0, 4))):

    g1=implicit_plot(y^2==4*c*x, (x,-5,5), (y, -7, 7))

    g2=implicit_plot(x==-c, (x,-5,5), (y, -7, 7))

    g3=implicit_plot(y*sqrt(4*c*x1)==2*c*(x+x1), (x,-5,5), (y, -7, 7))

    g4=implicit_plot(y*(-sqrt(4*c*x1))==2*c*(x+x1), (x,-5,5), (y, -7, 7))

    show(g1+g2+g3+g4, axes=True, frame=False, aspect_ratio=0.5)
```


c
0.992
x1
1.0
6
4
2
-2
-4
-6
-4
-2
2
4

30-11-2016

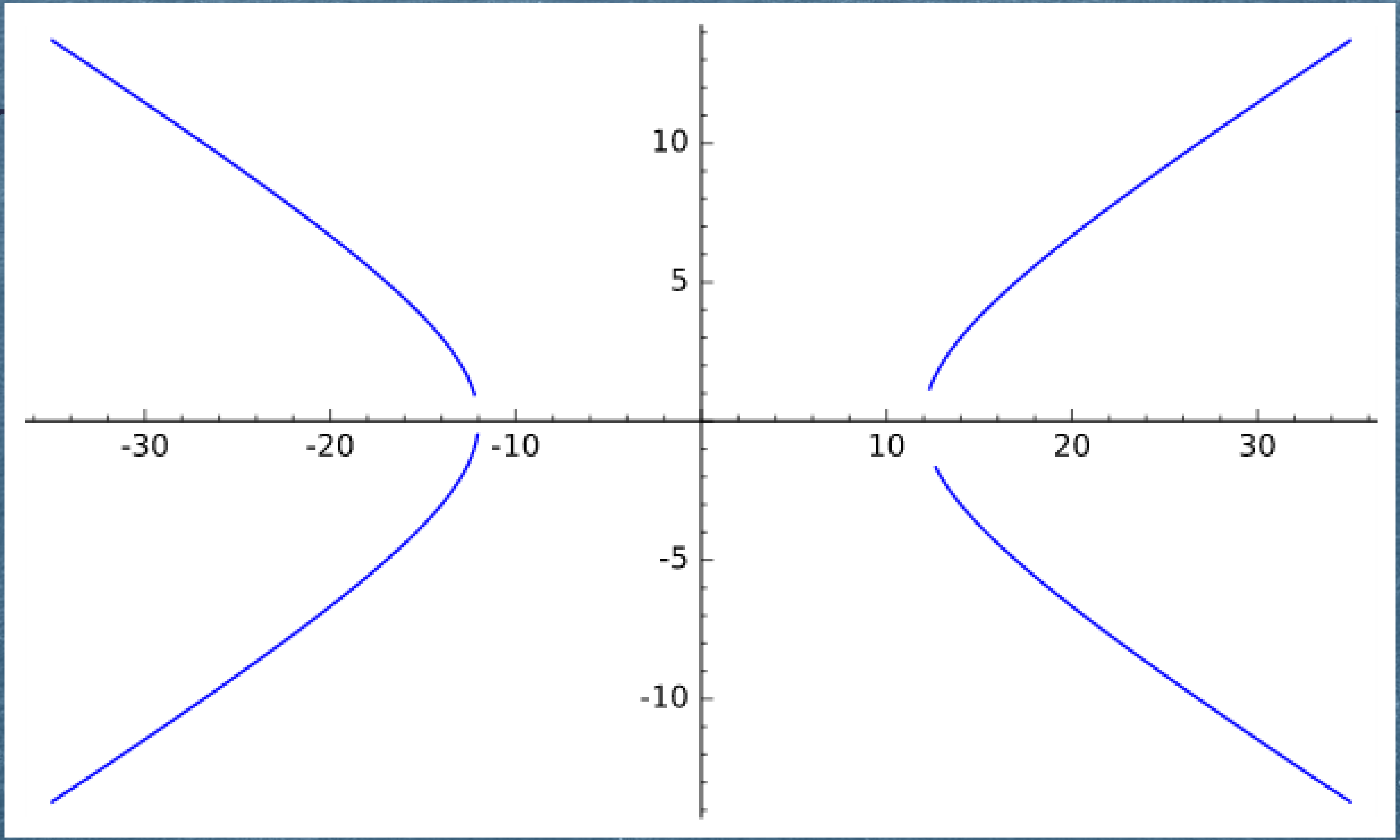
Χρήστος Τσαγγάρης
ΕΔΙΠ Τμήματος Μαθηματικών

Υπερβολή

Άσκηση 1η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{144} - \frac{y^2}{25} = 1.$

Λύση 1η:

plot((sqrt(25*(x^2/144-1)), -sqrt(25*(x^2/144-1))), (x, -35, 35), aspect_ratio=1.5)

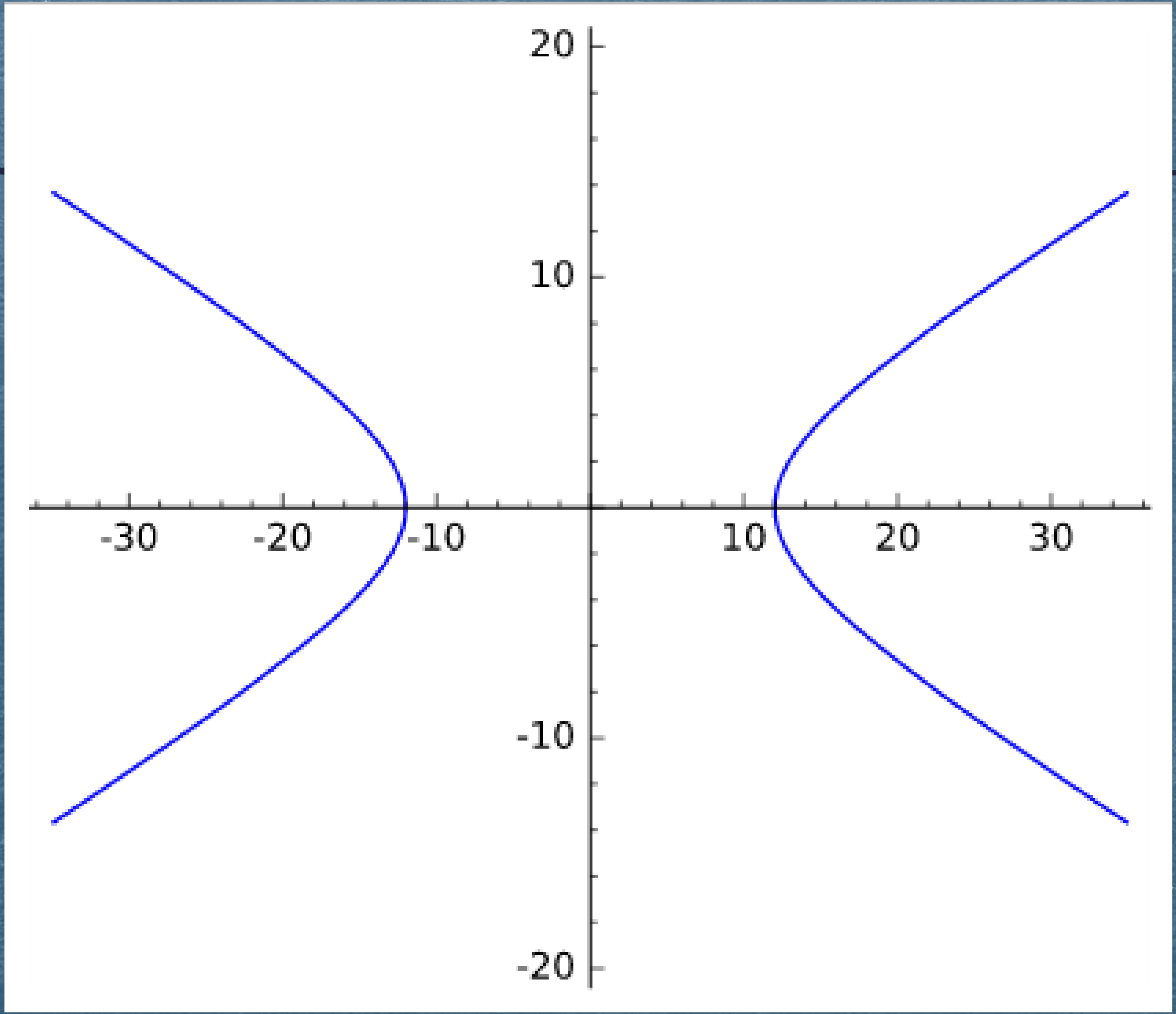


10
5
-5
-10
-30
-20
-10
10
20
30

Άσκηση 1η: Συνέχεια …

Λύση 2η:

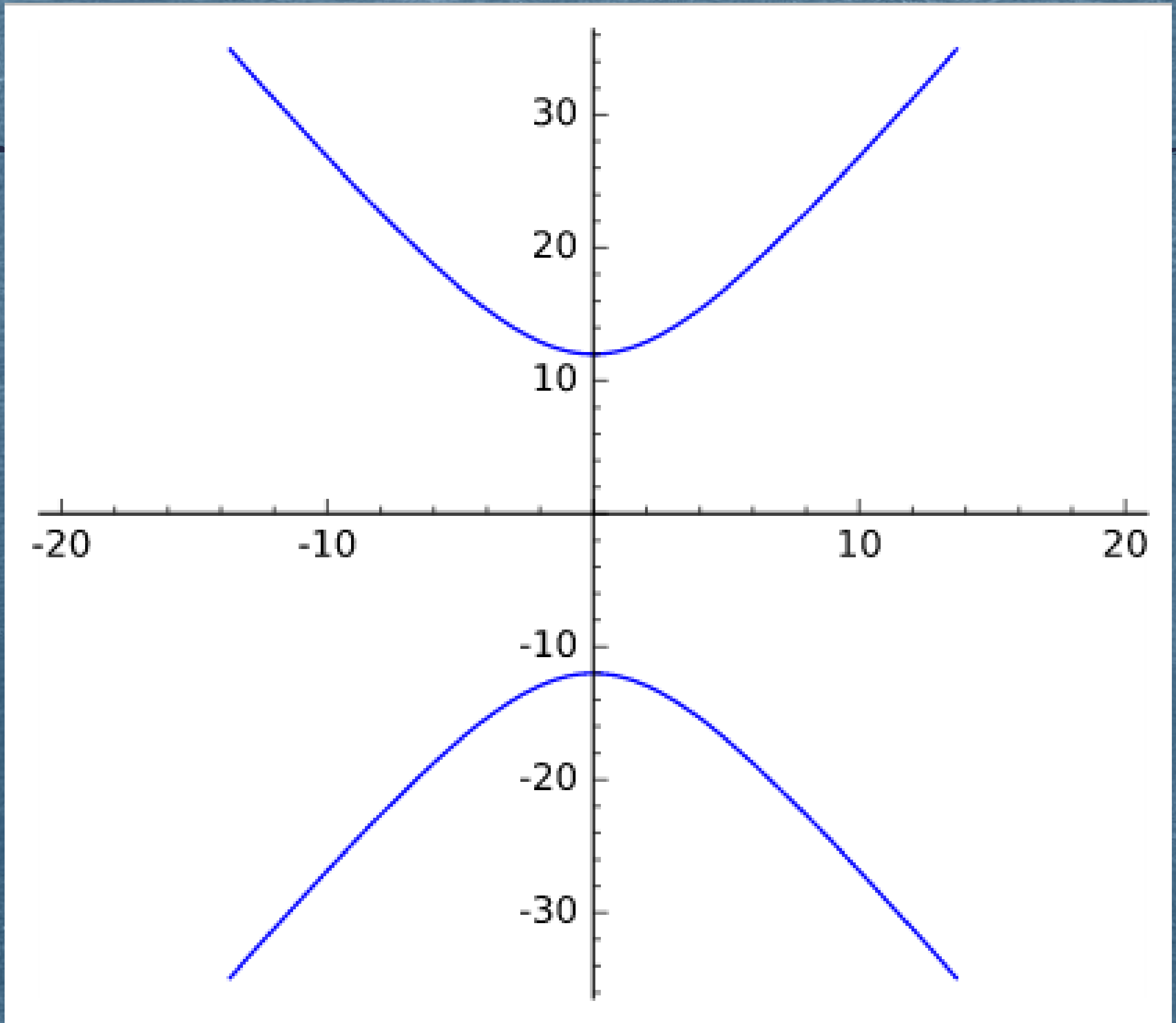
```
y=var('y')

implicit_plot(x^2/144 - y^2/25 == 1, (x, -35, 35), (y, -20, 20),
                                      axes=True, frame=False, aspect_ratio=1.5)
```


20
10
-10
-20
-30
-20
-10
10
20
30

Άσκηση 2η: Να σχεδιάσετε την υπερβολή $\frac{y^2}{144} - \frac{x^2}{25} = 1$.

Λύση:

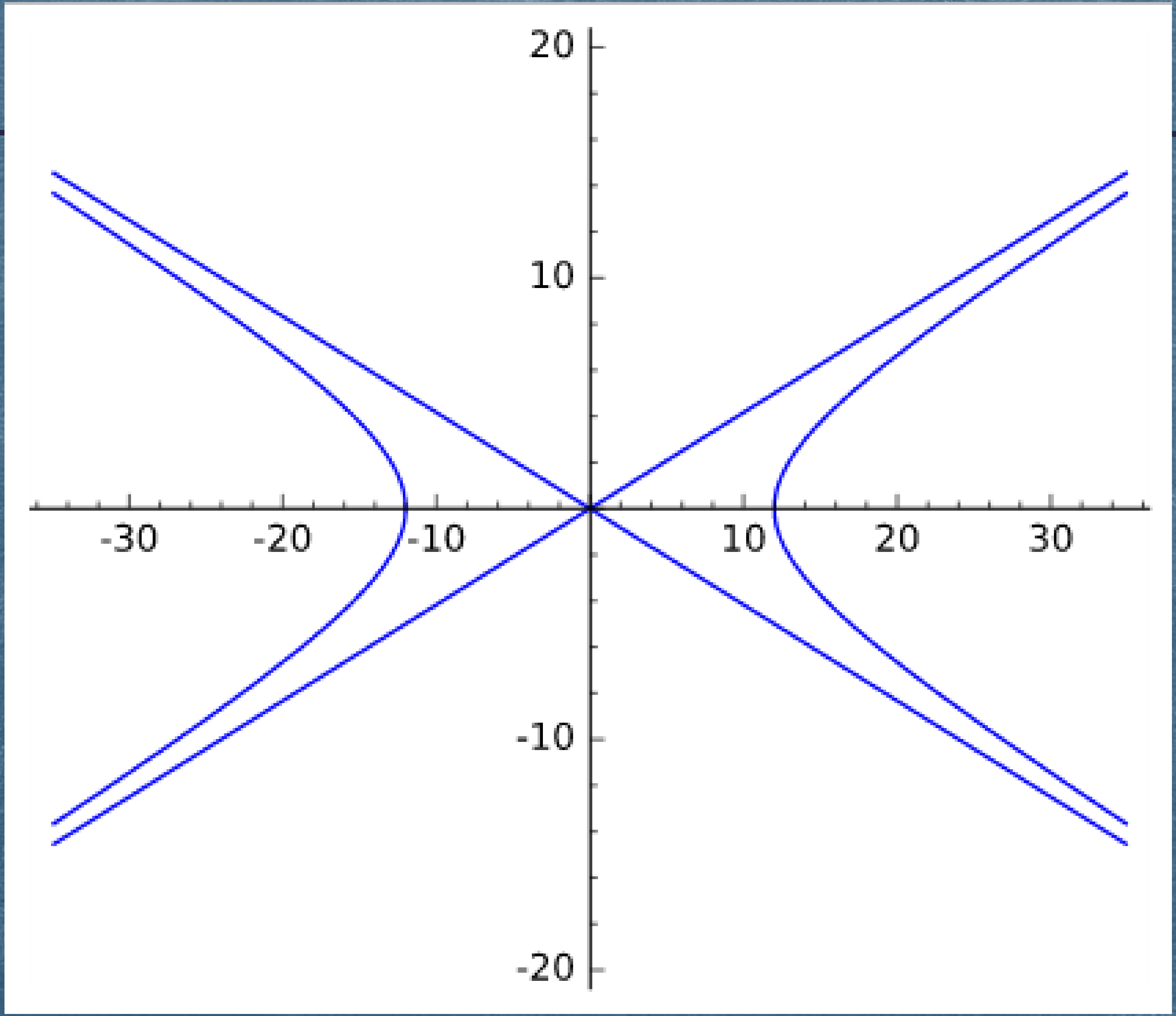
```
implicit_plot(y^2/144 - x^2/25 == 1, (x, -20, 20), (y, -35, 35),
                                   axes=True, frame=False, aspect_ratio=0.5)
```


30
20
10
-20
-10
10
20
-10
-20
-30

Άσκηση 3η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{144} - \frac{y^2}{25} = 1$ μαζί με τις ασύμπτωτές της.

Λύση:

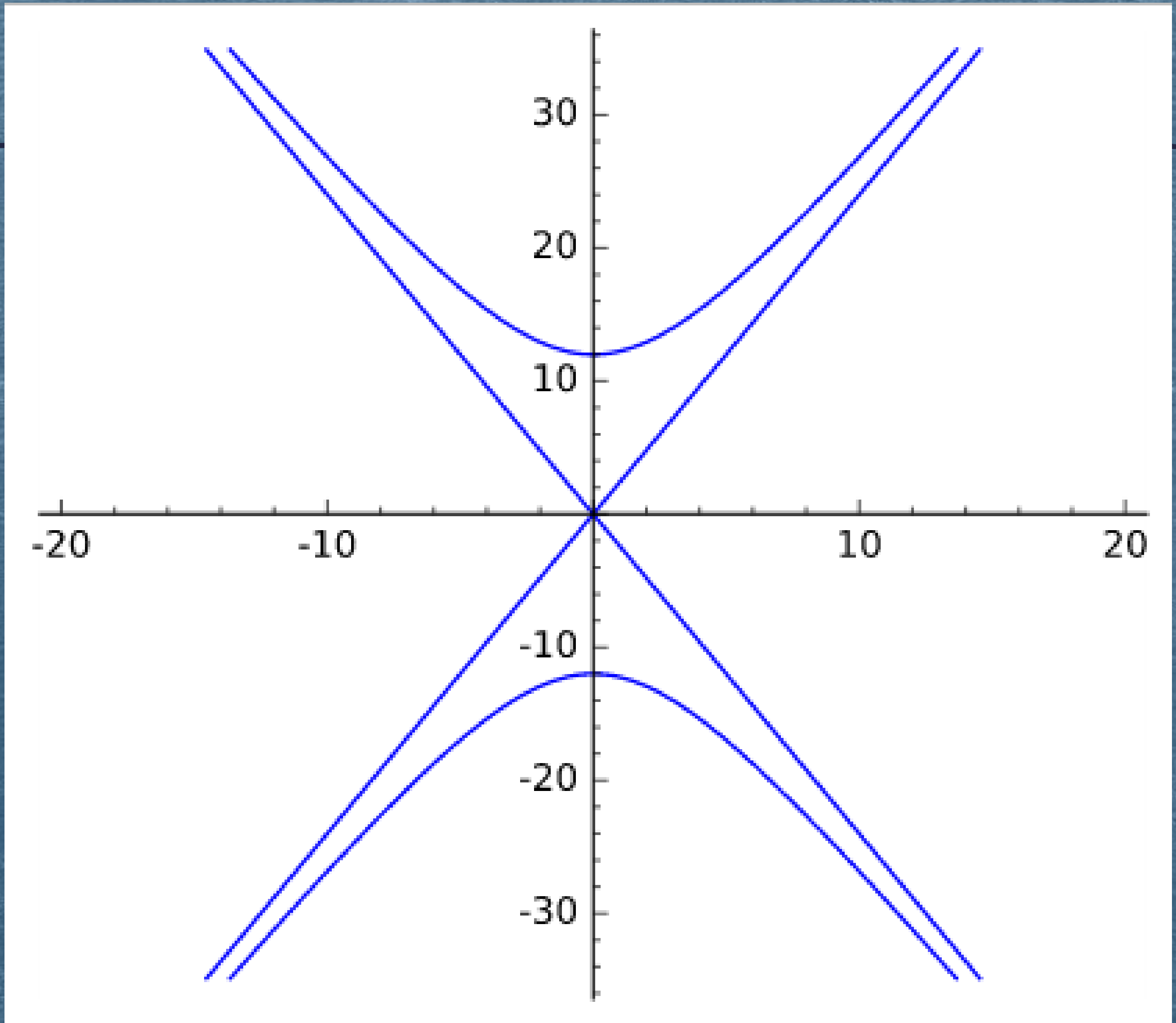
```
g1=implicit_plot(x^2/144 - y^2/25 == 1, (x, -35, 35), (y, -20, 20))

g2=implicit_plot(y==5/12*x, (x, -35, 35), (y, -20, 20))

g3=implicit_plot(y==-5/12*x, (x, -35, 35), (y, -20, 20))

show(g1+g2+g3,axes=True, frame=False, aspect_ratio=1.5)
```


20
10
-10
-20
-30
-20
-10
10
20
30

Άσκηση 4η: Να σχεδιάσετε την υπερβολή $\frac{y^2}{144} - \frac{x^2}{25} = 1$ μαζί με τις ασύμπτωτές της.

Λύση:

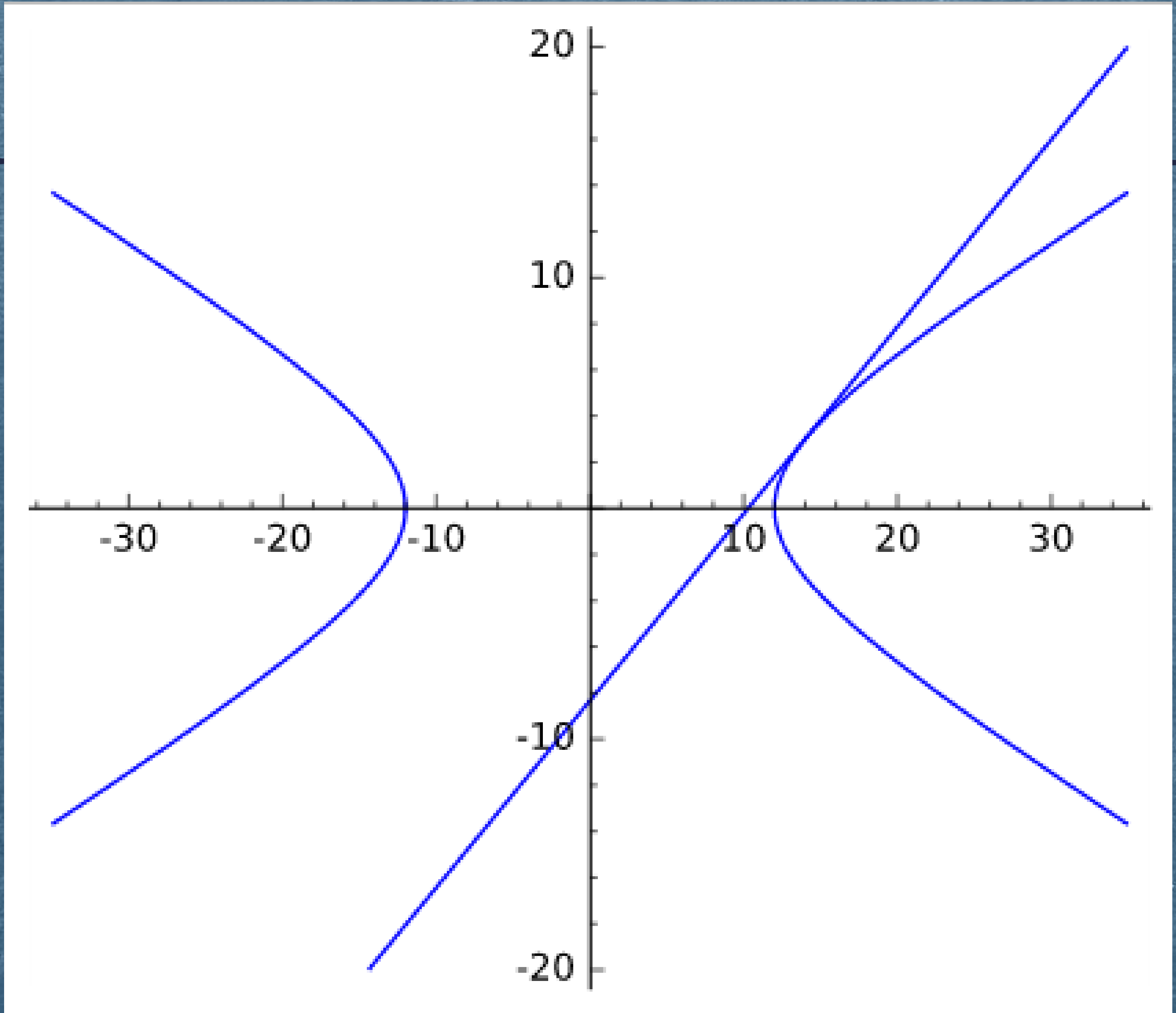
```
g1=implicit_plot(y^2/144 - x^2/25 == 1, (x, -20, 20), (y, -35, 35))

g2=implicit_plot(y==12/5*x, (x, -20, 20), (y, -35, 35))

g3=implicit_plot(y==-12/5*x, (x, -20, 20), (y, -35, 35))

show(g1+g2+g3, axes=True, frame=False, aspect_ratio=0.5)
```


30
20
10
-10
-20
-30
-20
-10
10
20

Άσκηση 5η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{144} - \frac{y^2}{25} = 1$ καθώς και την εφαπτομένη στο σημείο (14, 3).

Λύση:

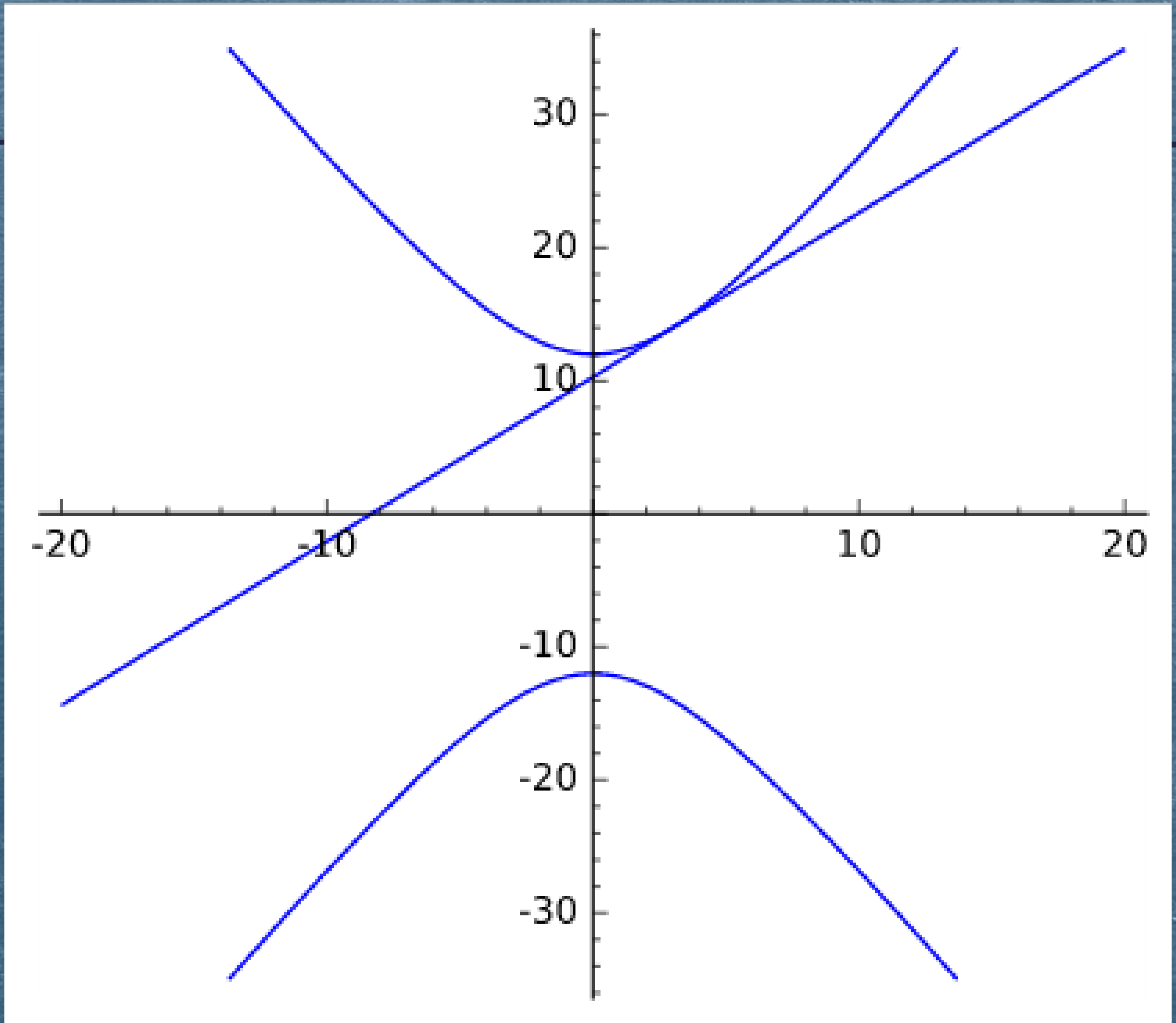
```
g1=implicit_plot(x^2/144 - y^2/25 == 1, (x, -35, 35), (y, -20, 20))

g2=implicit_plot(14*x/144 - 3*y/25 == 1, (x, -35, 35), (y, -20, 20))

show(g1+g2,axes=True, frame=False, aspect_ratio=1.5)
```


20
10
-10
-20
-30
-20
-10
10
20
30

Άσκηση 6η: Να σχεδιάσετε την υπερβολή $\frac{y^2}{144} - \frac{x^2}{25} = 1$ καθώς και την εφαπτομένη στο σημείο (3, 14).

Λύση:

```
g1=implicit_plot(y^2/144 - x^2/25 == 1, (x, -20, 20), (y, -35, 35))

g2=implicit_plot(14*y/144 - 3*x/25 == 1, (x, -20, 20), (y, -35, 35))

show(g1+g2, axes=True, frame=False, aspect_ratio=0.5)
```

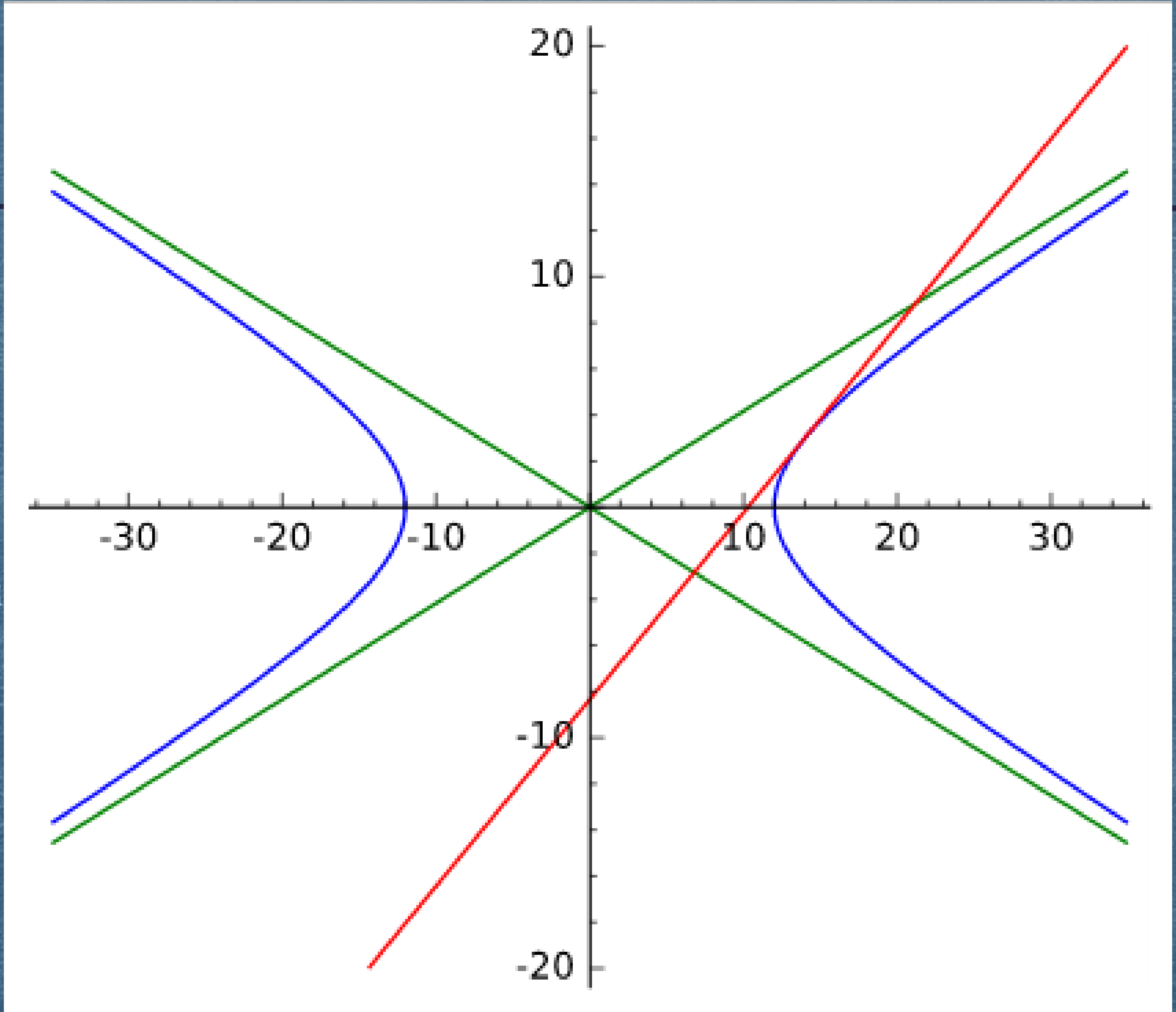


30
20
10
-20
-10
10
20
-10
-20
-30

Άσκηση 7η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{144} - \frac{y^2}{25} = 1$ μαζί με τις ασύμπτωτές της, καθώς και την εφαπτομένη της στο σημείο (14, 3).

Λύση:

```
g1=implicit_plot(x^2/144 - y^2/25 == 1, (x, -35, 35), (y, -20, 20))
g2=implicit_plot(y==5/12*x, (x, -35, 35), (y, -20, 20), color='green')
g3=implicit_plot(y==-5/12*x, (x, -35, 35), (y, -20, 20), color='green')
g4=implicit_plot(14*x/144 - 3*y/25 == 1, (x, -35, 35), (y, -20, 20), color='red')
show(g1+g2+g3+g4,axes=True, frame=False, aspect_ratio=1.5)
```

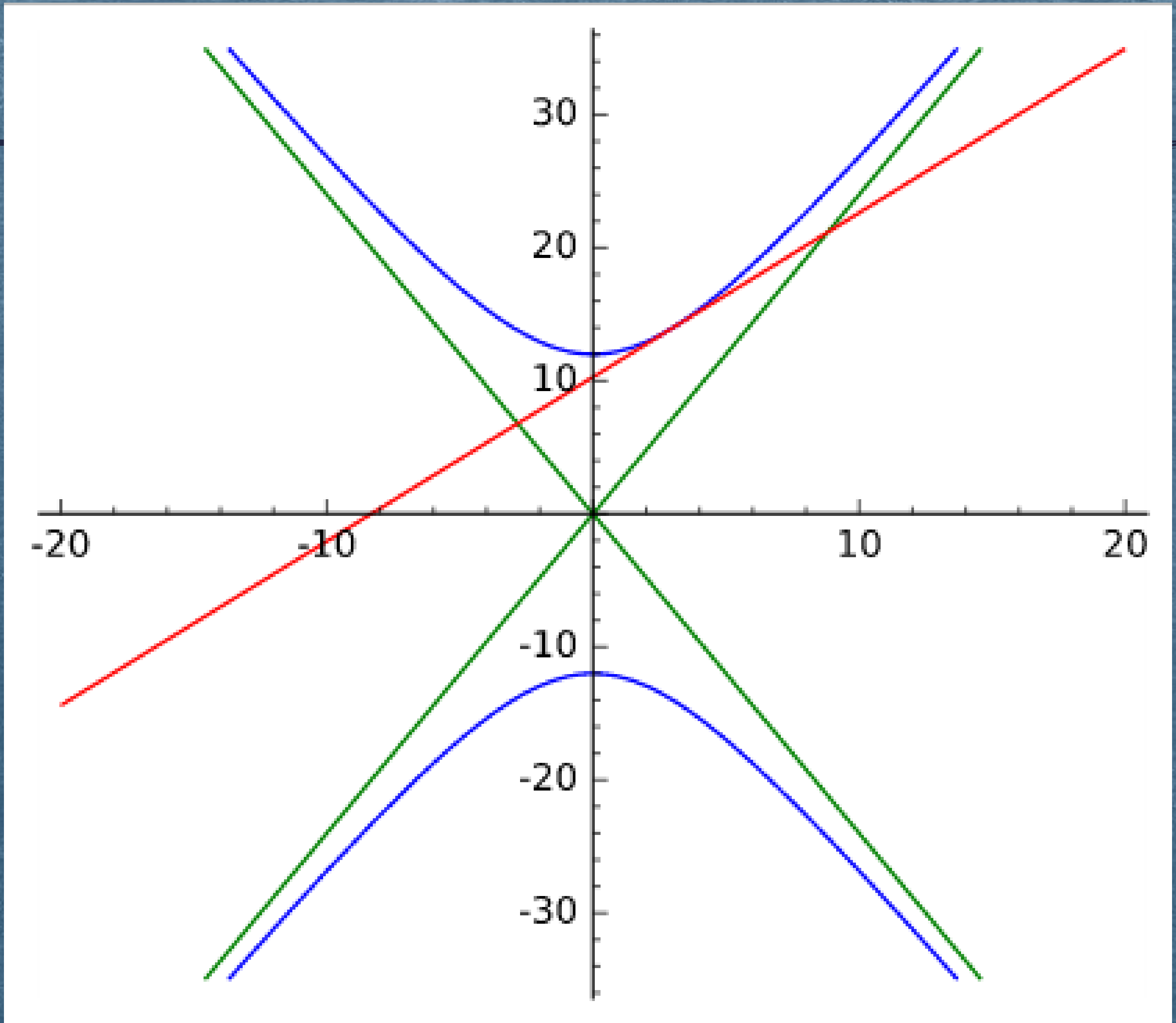


20
10
-10
-20
-30
-20
-10
10
20
30

Άσκηση 8η: Να σχεδιάσετε την υπερβολή $\frac{y^2}{144} - \frac{x^2}{25} = 1$ μαζί με τις ασύμπτωτές της, καθώς και την εφαπτομένη της στο σημείο (3, 14).

Λύση:

```
g1=implicit_plot(y^2/144 - x^2/25 == 1, (x, -20, 20), (y, -35, 35))

g2=implicit_plot(y==12/5*x, (x, -20, 20), (y, -35, 35), color='green')

g3=implicit_plot(y==-12/5*x, (x, -20, 20), (y, -35, 35), color='green')

g4=implicit_plot(14*y/144 - 3*x/25 == 1, (x, -20, 20), (y, -35, 35), color='red')

show(g1+g2+g3+g4, axes=True, frame=False, aspect_ratio=0.5)
```

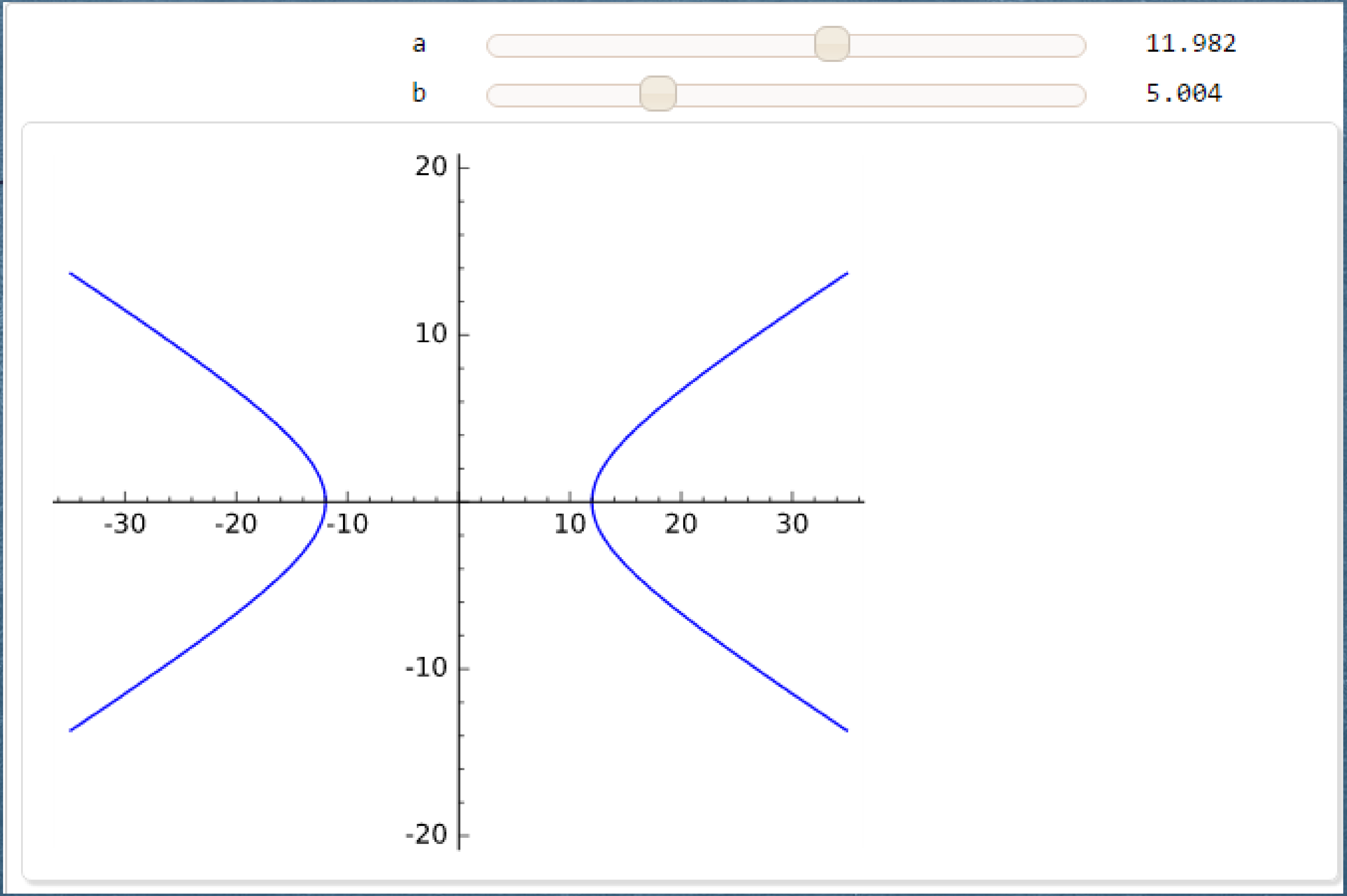


30
20
10
-10
-20
-30
-20
-10
10
20

Άσκηση 9η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ για διάφορες τιμές των παραμέτρων a και b.

Λύση:

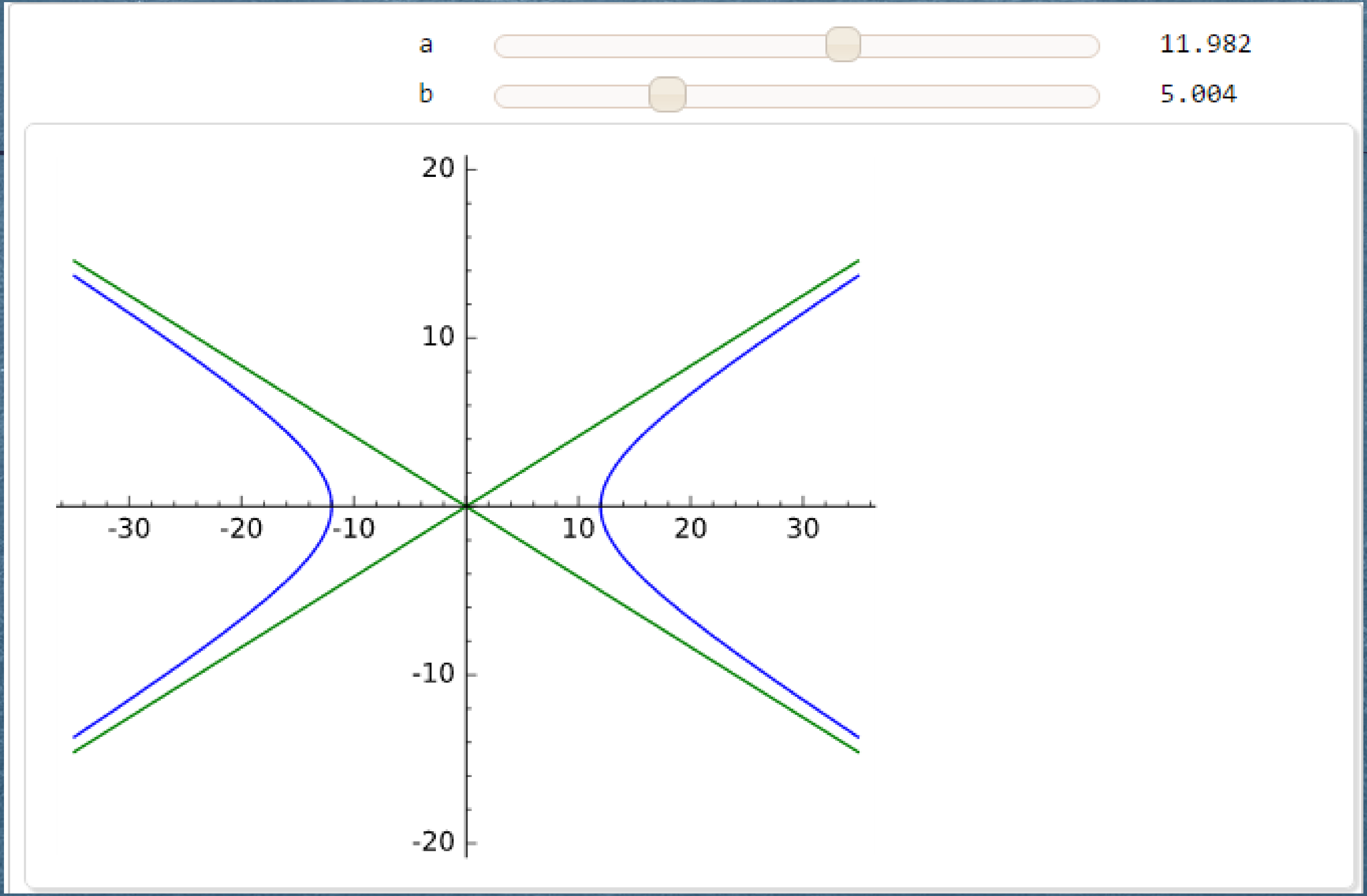
```
@interact
def my_plot(a=(12, (1, 20)), b=(5, (1, 15))):
    g1=implicit_plot(x^2/a^2 - y^2/b^2 == 1, (x, -35, 35), (y, -20, 20))
    show(g1, axes=True, frame=False, aspect_ratio=1.5)
```


a
11.982
b
5.004
20
10
-10
-20
-30
-20
-10
10
20
30

Άσκηση 10η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ μαζί με τις ασύμπτωτές της για διάφορες τιμές των παραμέτρων α και b.

Λύση:

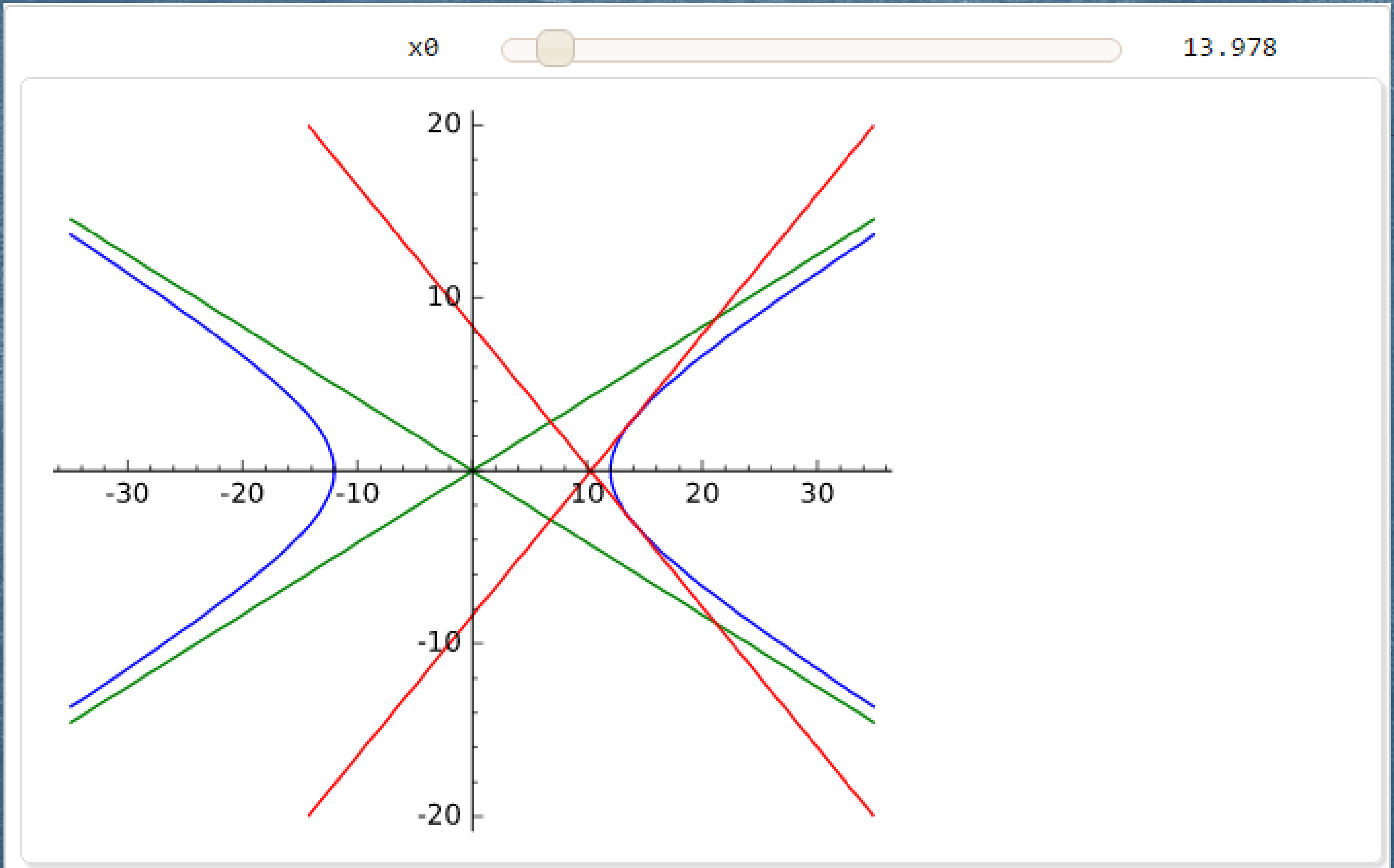
```
@interact
def my_plot(a=(12, (1, 20)), b=(5, (1, 15))):
    g1=implicit_plot(x^2/a^2 - y^2/b^2 == 1, (x, -35, 35), (y, -20, 20))
    g2=implicit_plot(y==b/a*x, (x, -35, 35), (y, -20, 20), color='green')
    g3=implicit_plot(y==-b/a*x, (x, -35, 35), (y, -20, 20), color='green')
    show(g1+g2+g3, axes=True, frame=False, aspect_ratio=1.5)
```


a
11.982
b
5.004
20
10
-10
-20
-30
-20
-10
10
20
30

Άσκηση 11η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{144} - \frac{y^2}{25} = 1$ μαζί με τις ασύμπτωτές της καθώς και τις εφαπτομένες της στα συμμετρικά σημεία ως προς τον άξονα xx'.

Λύση:

```
@interact
def my_plot(x0=(14, (12, 35))):
    g1=implicit_plot(x^2/144 - y^2/25 == 1, (x, -35, 35), (y, -20, 20))
    g2=implicit_plot(y==5/12*x, (x, -35, 35), (y, -20, 20), color='green')
    g3=implicit_plot(y==-5/12*x, (x, -35, 35), (y, -20, 20), color='green')
    g4=implicit_plot(x0*x/144 - sqrt(25*(x0^2/144-1))*y/25 == 1, (x, -35, 35),
                                                  (y, -20, 20), color='red')
    g5=implicit_plot(x0*x/144 + sqrt(25*(x0^2/144-1))*y/25 == 1, (x, -35, 35),
                                                  (y, -20, 20), color='red')
    show(g1+g2+g3+g4+g5, axes=True, frame=False, aspect_ratio=1.5)
```

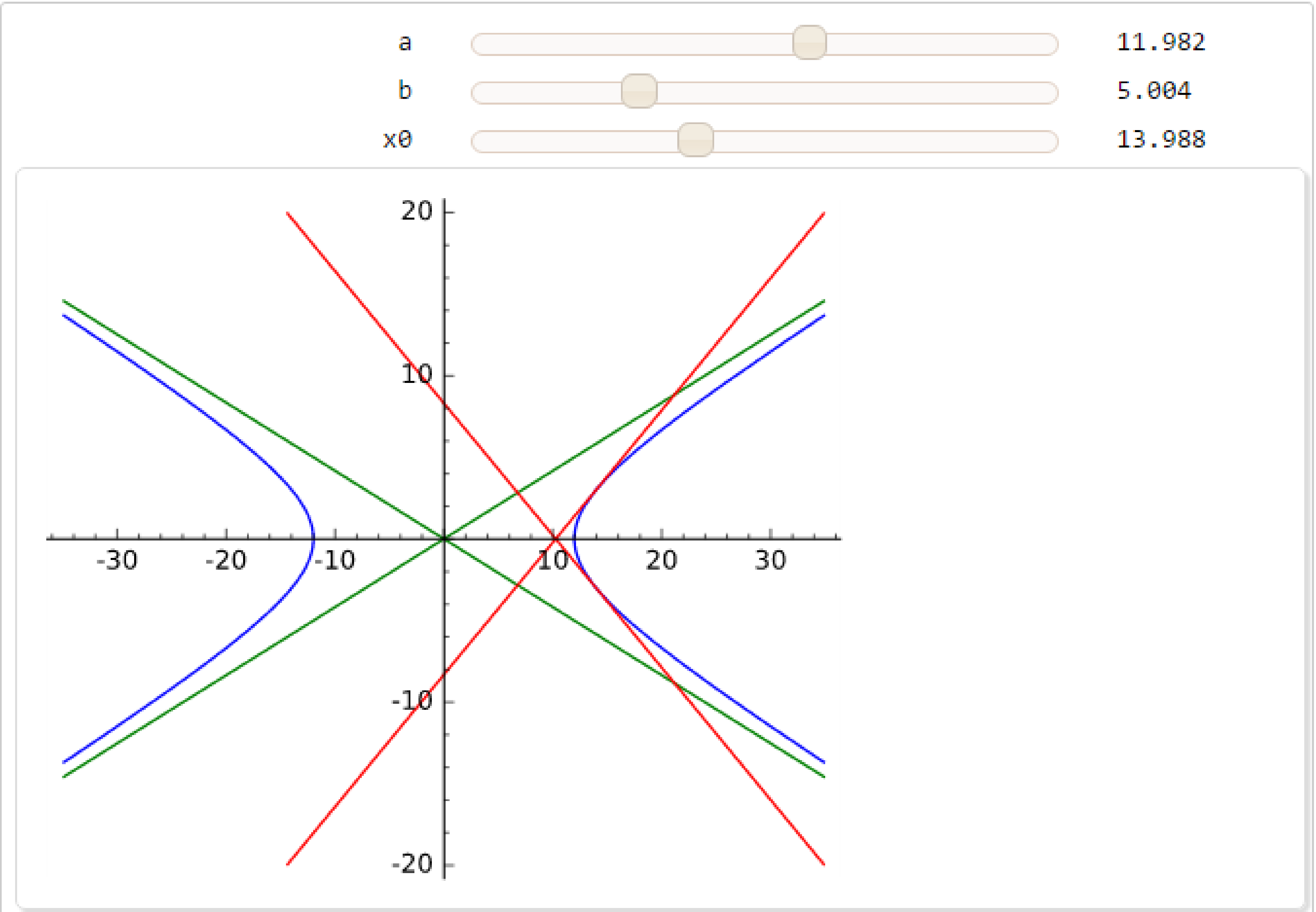


x0
13.978
20
10
-10
-20
-30
-20
-10
10
20
30

Άσκηση 12η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ μαζί με τις ασύμπτωτές της καθώς και τις εφαπτομένες της στα συμμετρικά σημεία ως προς τον άξονα xx', για διάφορες τιμές των παραμέτρων a, b και x0.

Λύση:

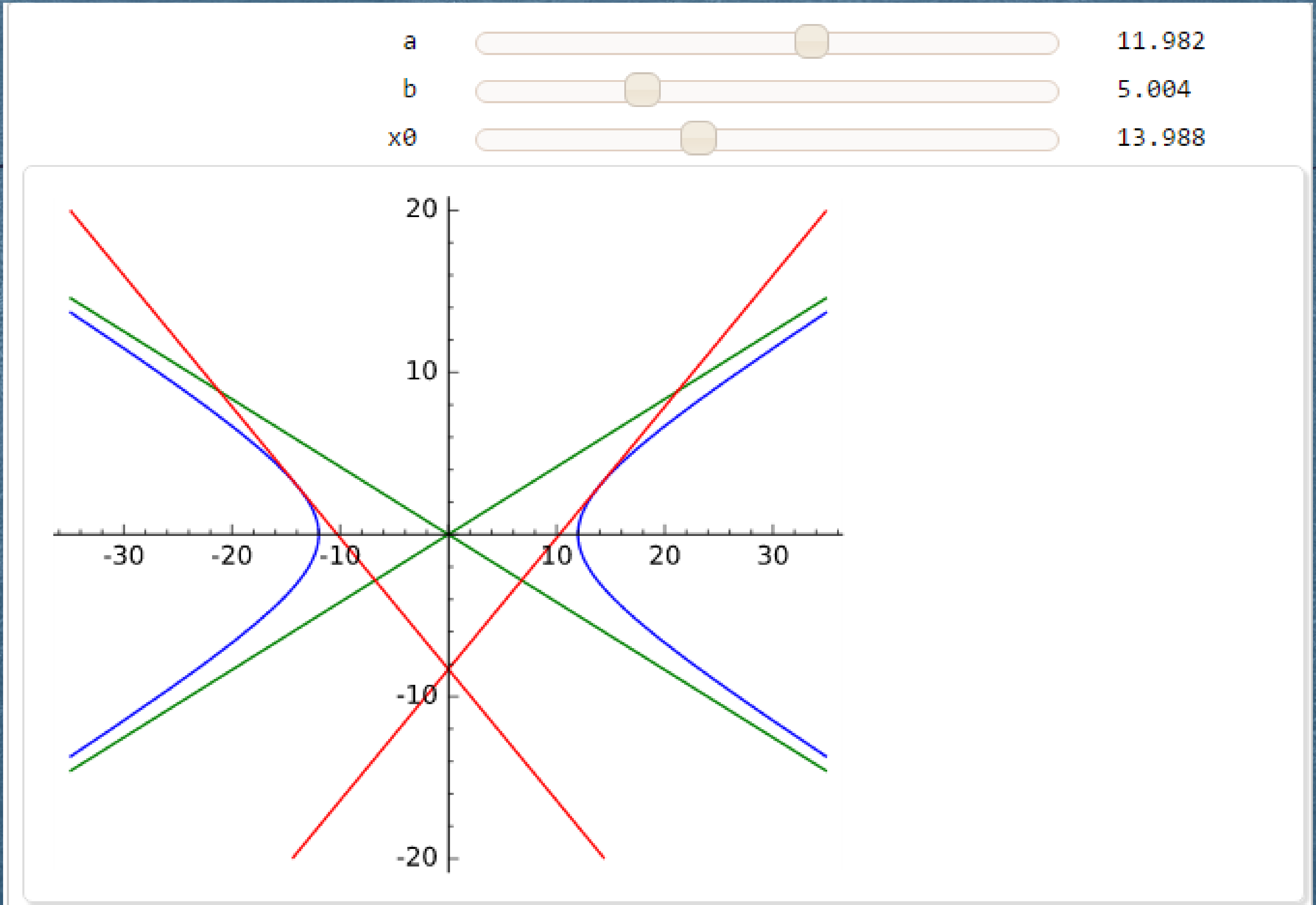
```
@interact
def my_plot(a=(12, (1, 20)), b=(5, (1, 15)), x0=(14, (1, 35))):
    if x0>a:
        g1=implicit_plot(x^2/a^2 - y^2/b^2 == 1, (x, -35, 35), (y, -20, 20))
        g2=implicit_plot(y==b/a*x, (x, -35, 35), (y, -20, 20), color='green')
        g3=implicit_plot(y==-b/a*x, (x, -35, 35), (y, -20, 20), color='green')
        g4=implicit_plot(x0*x/a^2 - sqrt(25*(x0^2/a^2-1))*y/b^2 == 1, (x, -35, 35),
                                    (y, -20, 20), color='red')
        g5=implicit_plot(x0*x/a^2 + sqrt(25*(x0^2/a^2-1))*y/b^2 == 1, (x, -35, 35),
                                    (y, -20, 20), color='red')
        show(g1+g2+g3+g4+g5, axes=True, frame=False, aspect_ratio=1.5)
```


a
11.982
b
5.004
x0
13.988
20
10
-10
-20
-30
-20
-10
10
20
30

Άσκηση 13η: Να σχεδιάσετε την υπερβολή $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ μαζί με τις ασύμπτωτές της καθώς και τις εφαπτομένες της στα συμμετρικά σημεία ως προς τον άξονα Oy', για διάφορες τιμές των παραμέτρων a, b και x0.

Λύση:

```
@interact
def my_plot(a=(12, (1, 20)), b=(5, (1, 15)), x0=(14, (1, 35))):
    if x0>a:
        g1=implicit_plot(x^2/a^2 - y^2/b^2 == 1, (x, -35, 35), (y, -20, 20))
        g2=implicit_plot(y==b/a*x, (x, -35, 35), (y, -20, 20), color='green')
        g3=implicit_plot(y==-b/a*x, (x, -35, 35), (y, -20, 20), color='green')
        g4=implicit_plot(x0*x/a^2 - sqrt(25*(x0^2/a^2-1))*y/b^2 == 1, (x, -35, 35),
                                                  (y, -20, 20), color='red')
        g5=implicit_plot(-x0*x/a^2 - sqrt(25*(x0^2/a^2-1))*y/b^2 == 1, (x, -35, 35),
                                                  (y, -20, 20), color='red')
        show(g1+g2+g3+g4+g5, axes=True, frame=False, aspect_ratio=1.5)
```


a 11.982
b 5.004
x0 13.988
20
10
-10
-20
-30
-20
-10
10
20
30

1-12-2016

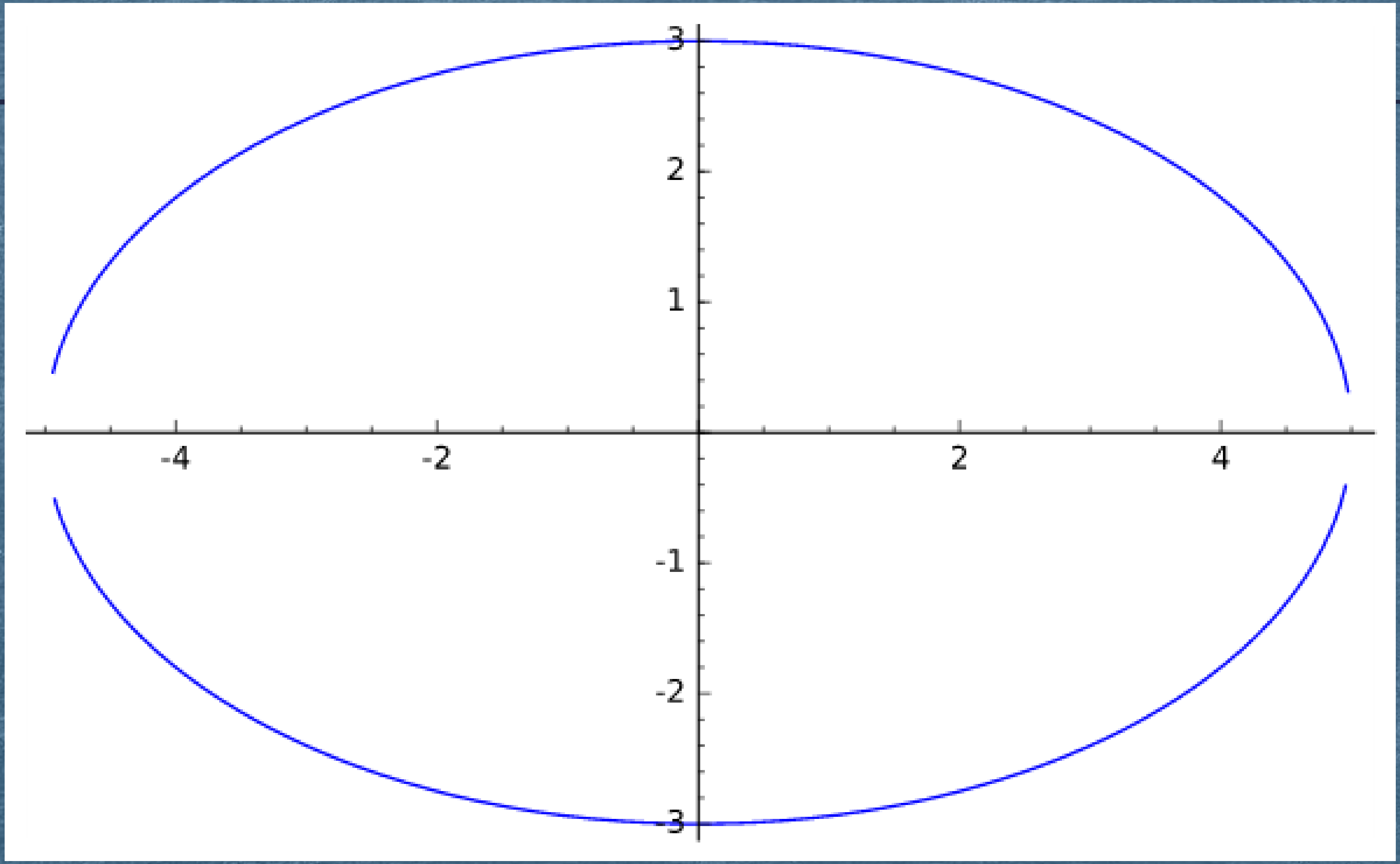
Χρήστος Τσαγγάρης
ΕΔΙΠ Τμήματος Μαθηματικών

Έλλειψη

Άσκηση 1η: Να σχεδιάσετε την έλλειψη $\frac{x^2}{25}+\frac{y^2}{9}=1.$

Λύση 1η:

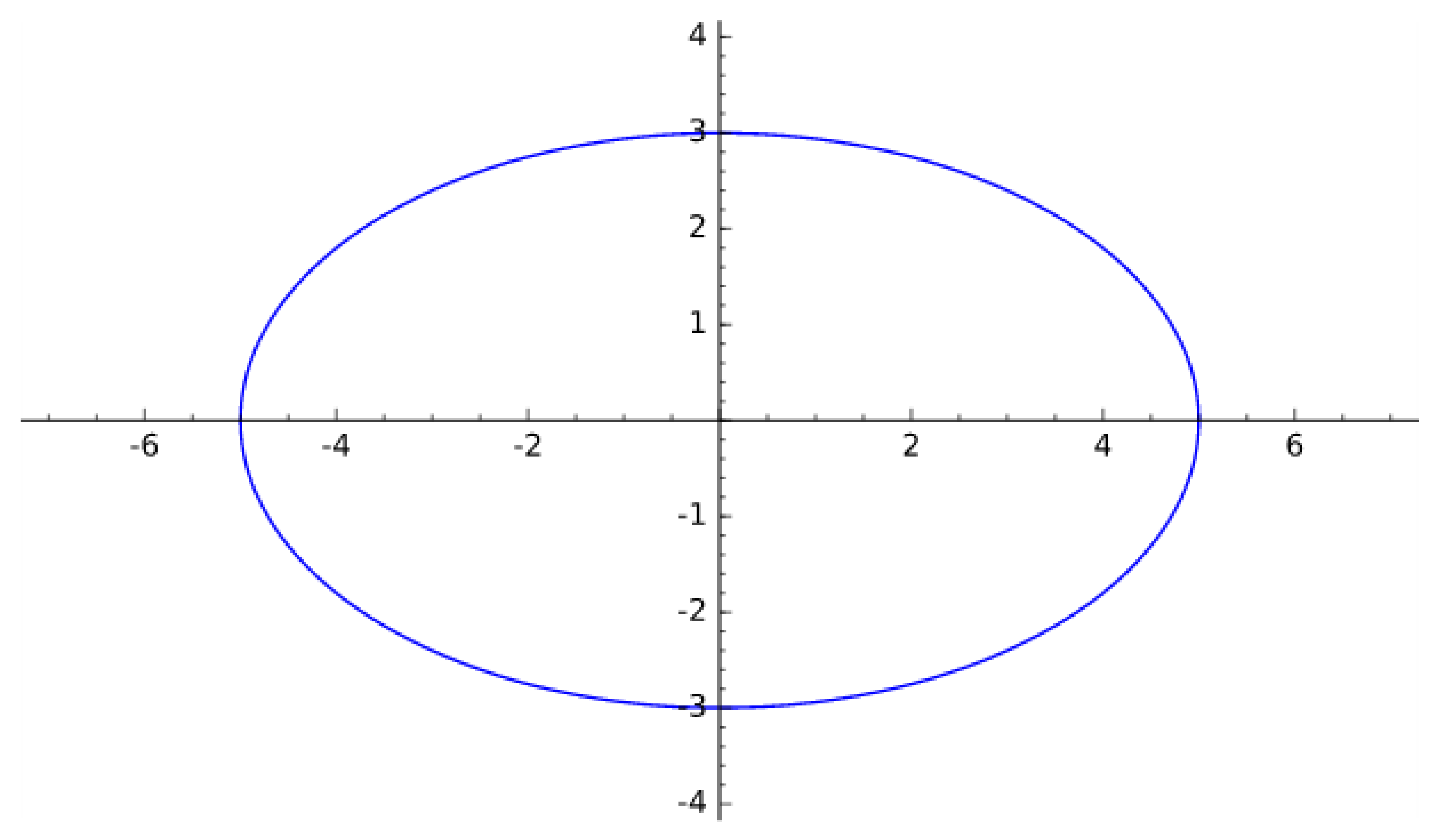
plot((sqrt(9*(1-x^2/25)), -sqrt(9*(1-x^2/25))), (x, -7, 7), aspect_ratio=1)


3
2
1
-1
-2
-3
-4
-2
2
4

Άσκηση 1η: Συνέχεια …

Λύση 2η:

```
y=var('y')

implicit_plot(x^2/25 + y^2/9 == 1, (x, -7, 7), (y, -4, 4),axes=True,
                                        frame=False, aspect_ratio=1)
```

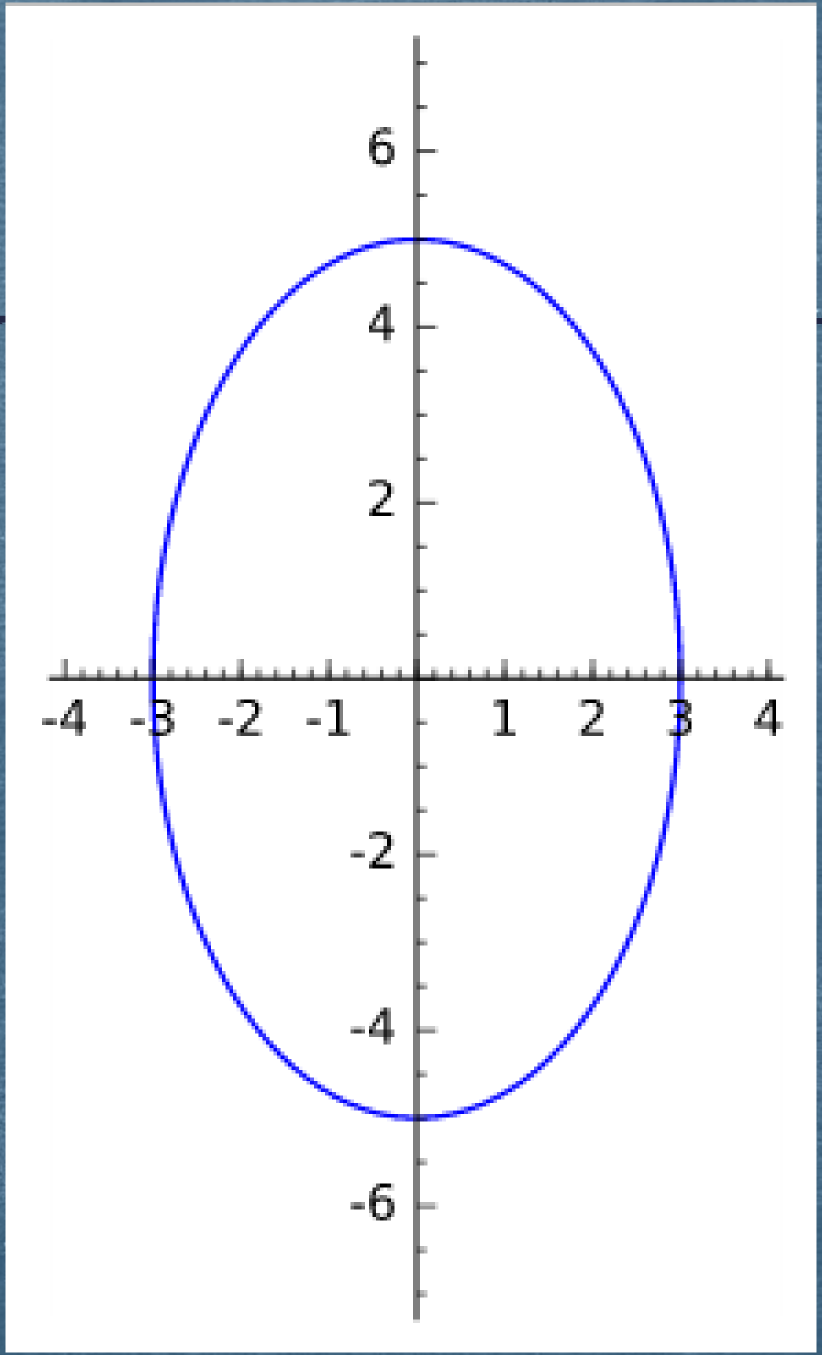


4
3
2
1
-6
-4
-2
2
4
6
-1
-2
-3
-4

Άσκηση 2η: Να σχεδιάσετε την έλλειψη $\frac{y^2}{25}+\frac{x^2}{9}=1.$

Λύση:

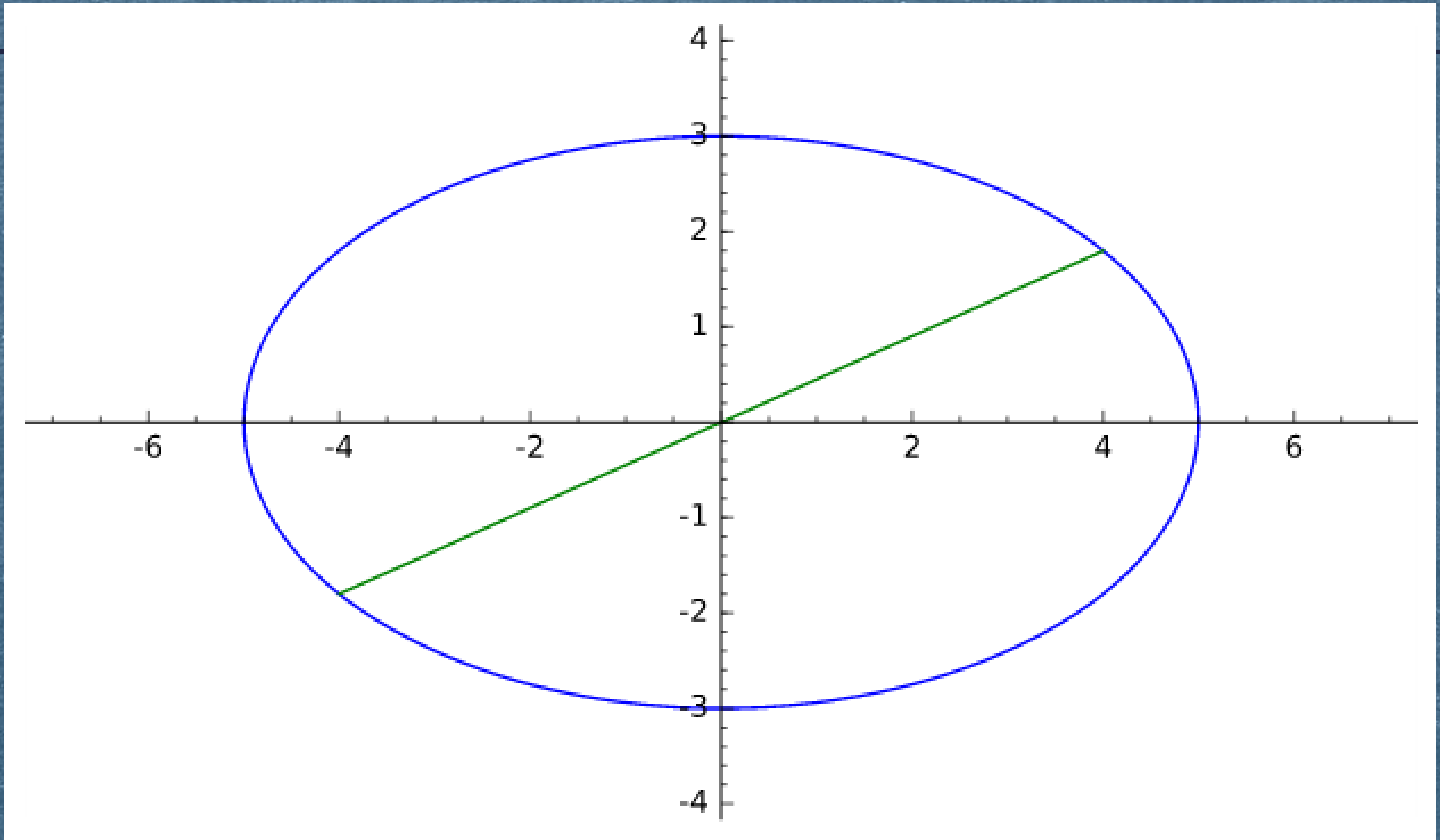
implicit_plot(y^2/25 + x^2/9 == 1, (x, -4, 4), (y, -7, 7), axes=True, frame=False, aspect_ratio=1)


6
4
2
-2
-4
-6
-4 -3 -2 -1 1 2 3 4

Άσκηση 3η: Να σχεδιάσετε την έλλειψη $\frac{x^2}{25}+\frac{y^2}{9}=1$ μαζί με τη διάμετρό της που περνάει από το σημείο (4, 9/5).

Λύση:

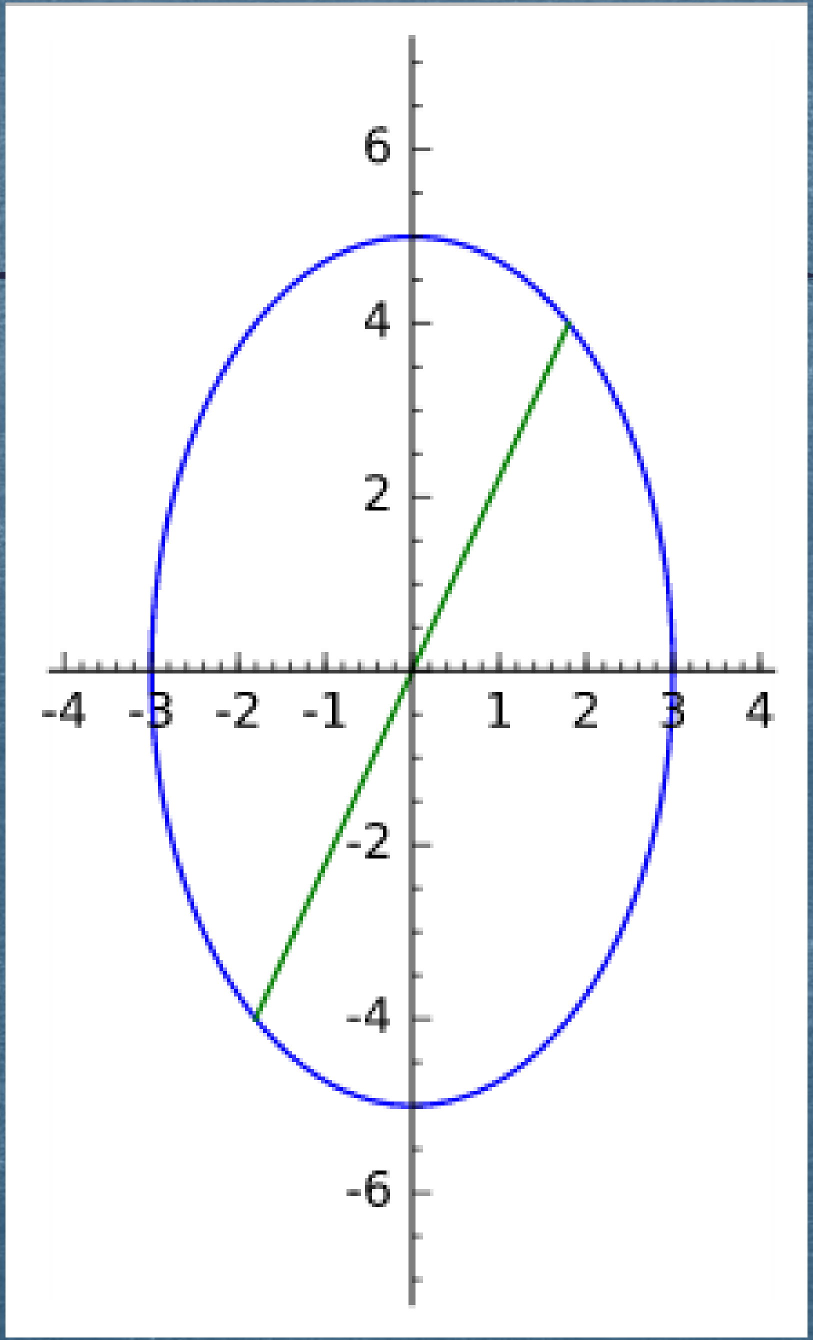
```
g1=implicit_plot(x^2/25 + y^2/9 == 1, (x, -7, 7), (y, -4, 4))

g2=line(((4, 9/5), (-4, -9/5)), color='green')

show(g1+g2,axes=True,frame=False, aspect_ratio=1)
```


4
3
2
1
-1
-2
-3
-4
-6
-4
-2
2
4
6

Άσκηση 4η: Να σχεδιάσετε την έλλειψη $\frac{y^2}{25}+\frac{x^2}{9}=1$ μαζί με τη διάμετρό της που περνάει από το σημείο (9/5, 4).

Λύση:

```
g1=implicit_plot(y^2/25 + x^2/9 == 1, (x, -4, 4), (y, -7, 7))

g2=line(((9/5, 4), (-9/5, -4)), color='green')

show(g1+g2,axes=True,frame=False, aspect_ratio=1)
```

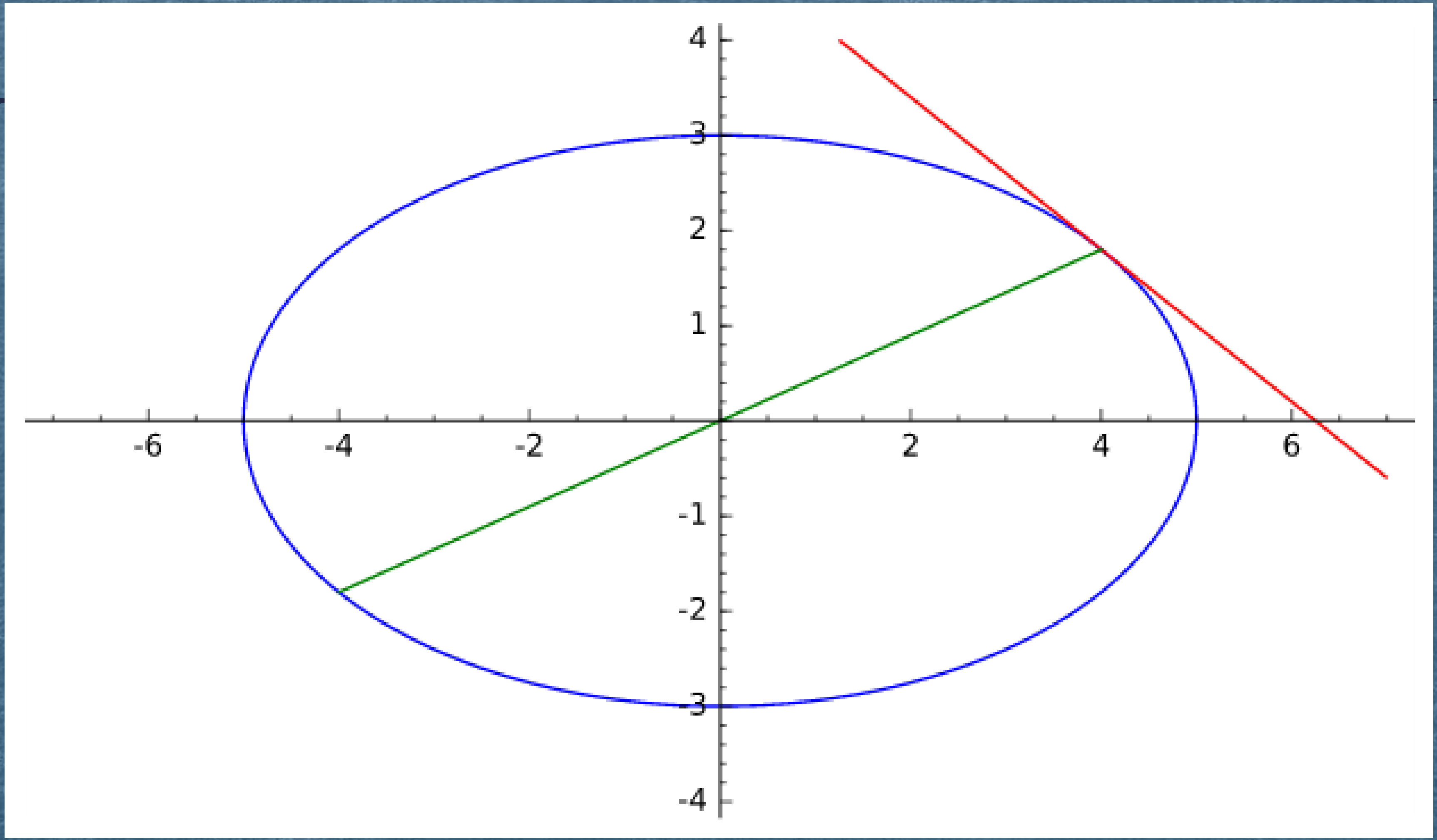


6
4
2
-2
-4
-6
-4 -3 -2 -1 1 2 3 4

Άσκηση 5η: Να σχεδιάσετε την έλλειψη $\frac{x^2}{25}+\frac{y^2}{9}=1$ μαζί με τη διάμετρό της που περνάει από το σημείο (4, 9/5) καθώς και την εφαπτομένη που διέρχεται από αυτό το σημείο.

Λύση:

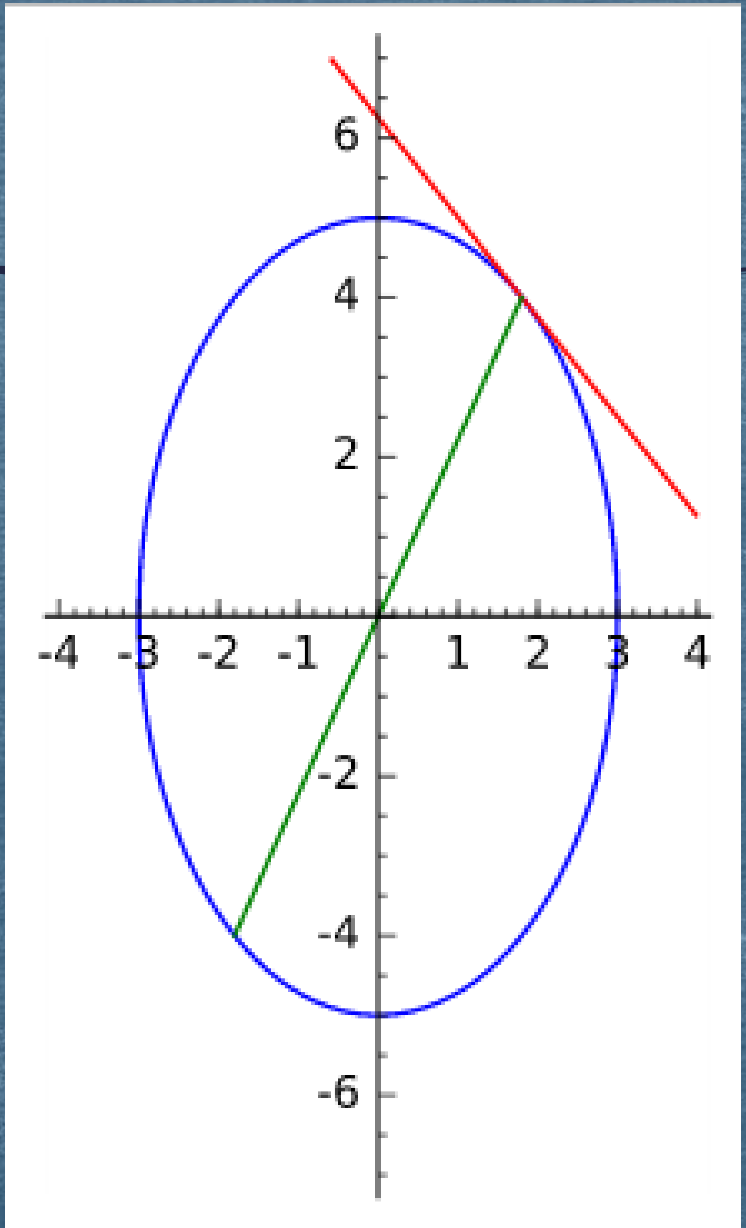
```
g1=implicit_plot(x^2/25 + y^2/9 == 1, (x, -7, 7), (y, -4, 4))

g2=line(((4, 9/5), (-4, -9/5)), color='green')

g3=implicit_plot(4*x/25 + (9/5)*y/9 == 1, (x, -7, 7), (y, -4, 4), color='red')

show(g1+g2+g3,axes=True,frame=False, aspect_ratio=1)
```


4
3
2
1
-1
-2
-3
-4
-6
-4
-2
2
4
6

Άσκηση 6η: Να σχεδιάσετε την έλλειψη $\frac{y^2}{25}+\frac{x^2}{9}=1$ μαζί με τη διάμετρό της που περνάει από το σημείο (9/5, 4) καθώς και την εφαπτομένη που διέρχεται από αυτό το σημείο.

Λύση:

```
g1=implicit_plot(y^2/25 + x^2/9 == 1, (x, -4, 4), (y, -7, 7))

g2=line(((9/5, 4), (-9/5, -4)), color='green')

g3=implicit_plot(4*y/25 + (9/5)*x/9 == 1, (x, -4, 4), (y, -7, 7), color='red')

show(g1+g2+g3,axes=True,frame=False, aspect_ratio=1)
```

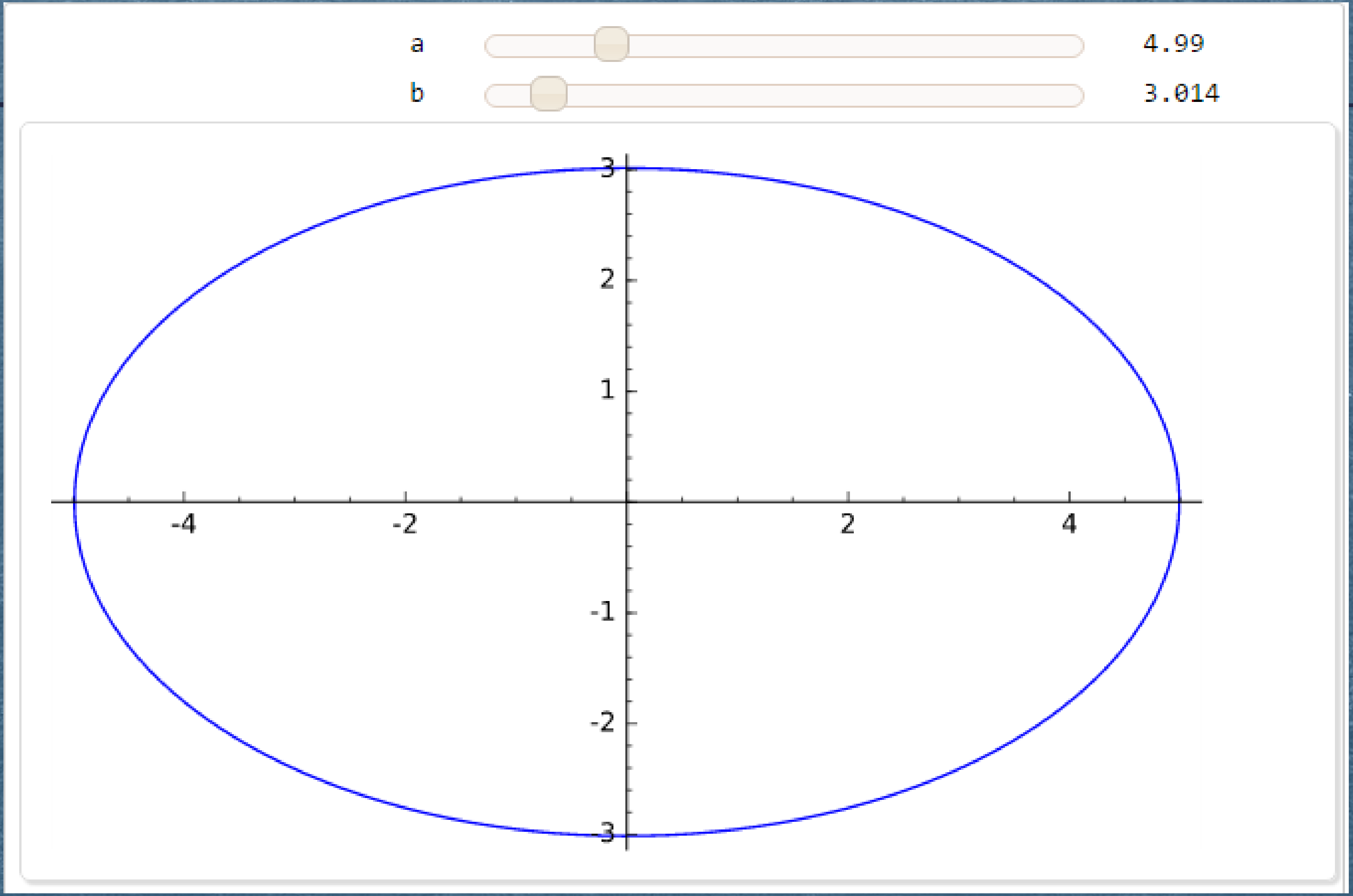


6
4
2
-2
-4
-6
-4 -3 -2 -1 1 2 3 4

Άσκηση 7η: Να σχεδιάσετε την έλλειψη $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$ για διάφορες τιμές των παραμέτρων a και b.

Λύση:

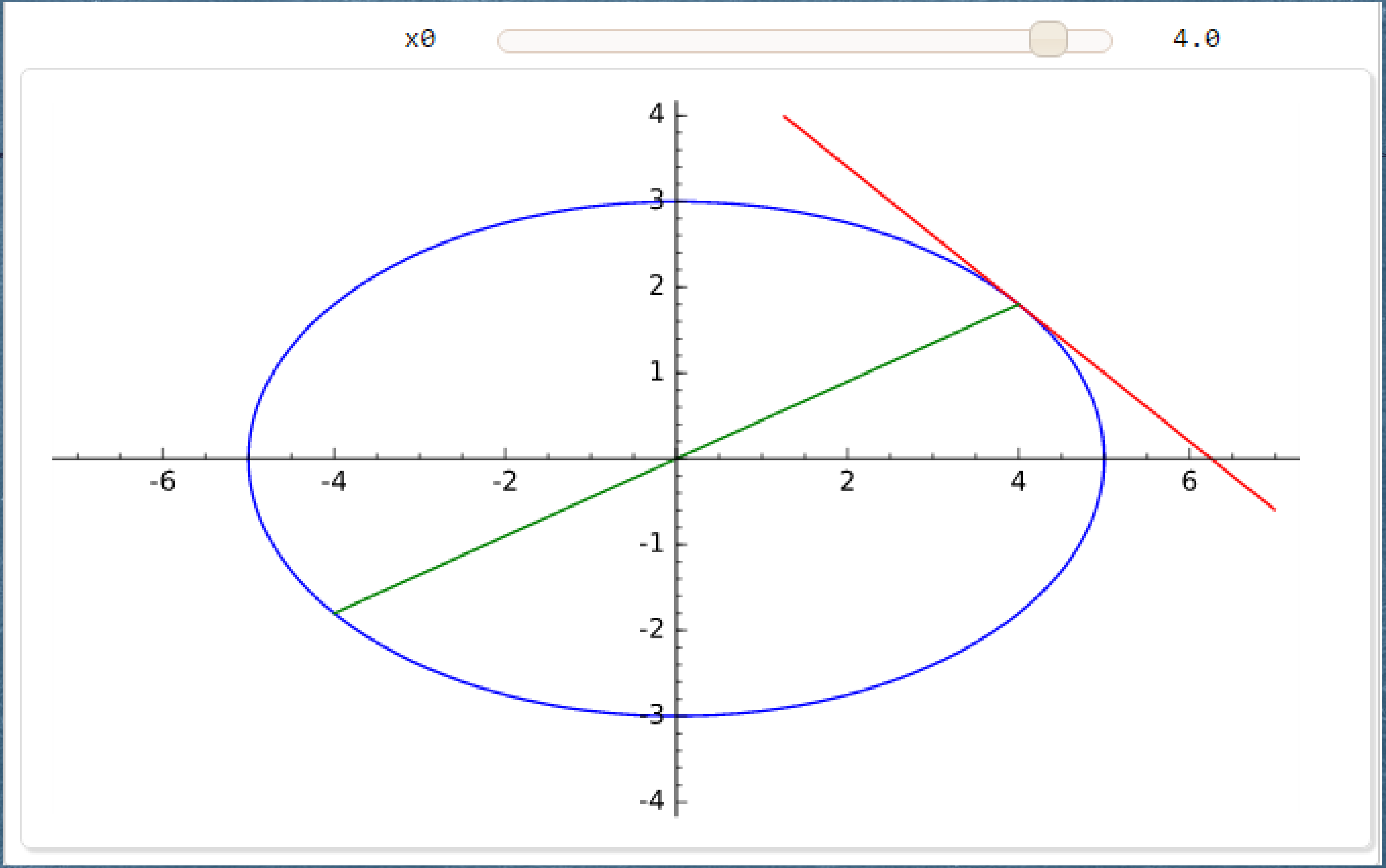
```
@interact
def my_plot(a=(5, (1, 20)), b=(3, (1, 20))):
    g1=implicit_plot(x^2/a^2 + y^2/b^2 == 1, (x, -a, a), (y, -b, b))
    show(g1, axes=True, frame=False, aspect_ratio=1)
```


a
4.99
b
3.014
3
2
1
-1
-2
-3
-4
-2
2
4

Άσκηση 8η: Να σχεδιάσετε την έλλειψη $\frac{x^2}{25}+\frac{y^2}{9}=1$ μαζί με την εφαπτόμενη της στα σημεία της που βρίσκονται στο 1ο και 2ο τεταρτημόριο , καθώς και τη διάμετρο που διέρχεται από αυτά.

Λύση:

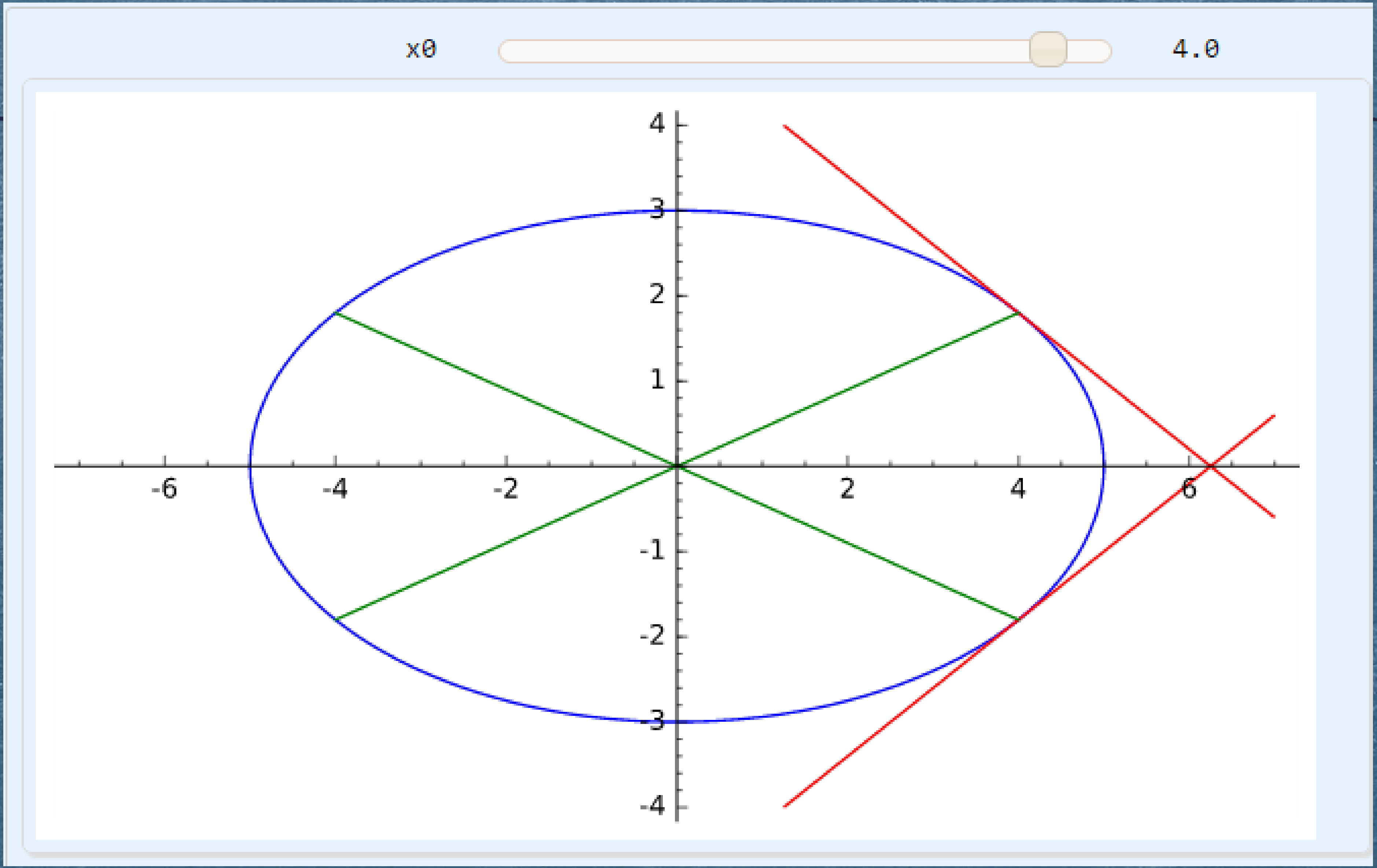
```
@interact
def my_plot(x0=(4, (-5,5))):
    g1=implicit_plot(x^2/25 + y^2/9 == 1, (x, -7, 7), (y, -4, 4))
    g2=line(((x0, sqrt(9*(1-x0^2/25))), (-x0, -sqrt(9*(1-x0^2/25)))), color='green')
    g3=implicit_plot(x0*x/25 + sqrt(9*(1-x0^2/25))*y/9 == 1, (x, -7, 7), (y, -4, 4),
                                                                color='red')
    show(g1+g2+g3,axes=True,frame=False, aspect_ratio=1)
```


x0
4.0
4
3
2
1
-1
-2
-3
-4
-6
-4
-2
2
4
6

Άσκηση 9η: Να σχεδιάσετε την έλλειψη $\frac{x^2}{25}+\frac{y^2}{9}=1$ μαζί με τις εφαπτομένες της στα συμμετρικά σημεία της, ως προς τον άξονα xx’ καθώς και τις διαμέτρους που διέρχονται από αυτά.

Λύση:

```
@interact
def my_plot(x0=(4, (-5,5))):
    g1=implicit_plot(x^2/25 + y^2/9 == 1, (x, -7, 7), (y, -4, 4))
    g2=line(((x0, sqrt(9*(1-x0^2/25))), (-x0, -sqrt(9*(1-x0^2/25)))), color='green')
    g3=line(((x0, -sqrt(9*(1-x0^2/25))), (-x0, +sqrt(9*(1-x0^2/25)))), color='green')
    g4=implicit_plot(x0*x/25 + sqrt(9*(1-x0^2/25))*y/9 == 1, (x, -7, 7), (y, -4, 4),
                                                                color='red')
    g5=implicit_plot(x0*x/25 - sqrt(9*(1-x0^2/25))*y/9 == 1, (x, -7, 7), (y, -4, 4),
                                                                color='red')
    show(g1+g2+g3+g4+g5,axes=True,frame=False, aspect_ratio=1)
```

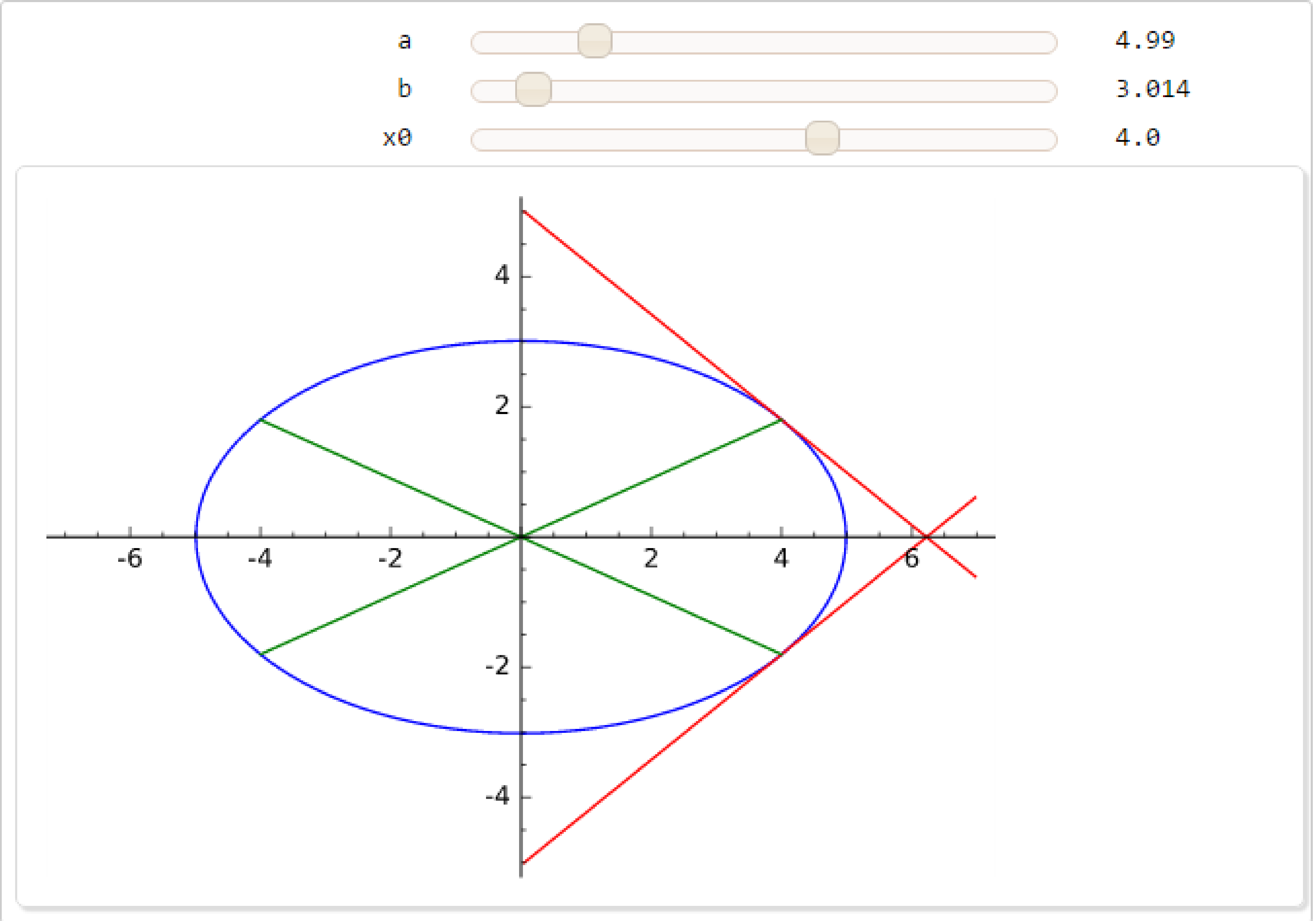


x0
4.0
4
3
2
1
-1
-2
-3
-4
-6
-4
-2
2
4
6

Άσκηση 10η: Να σχεδιάσετε την έλλειψη $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$ μαζί με τις εφαπτομένες της στα συμμετρικά σημεία της, ως προς τον άξονα xx' καθώς και τις διαμέτρους που διέρχονται από αυτά, για διάφορες τιμές των παραμέτρων a, b και x0.

Λύση:

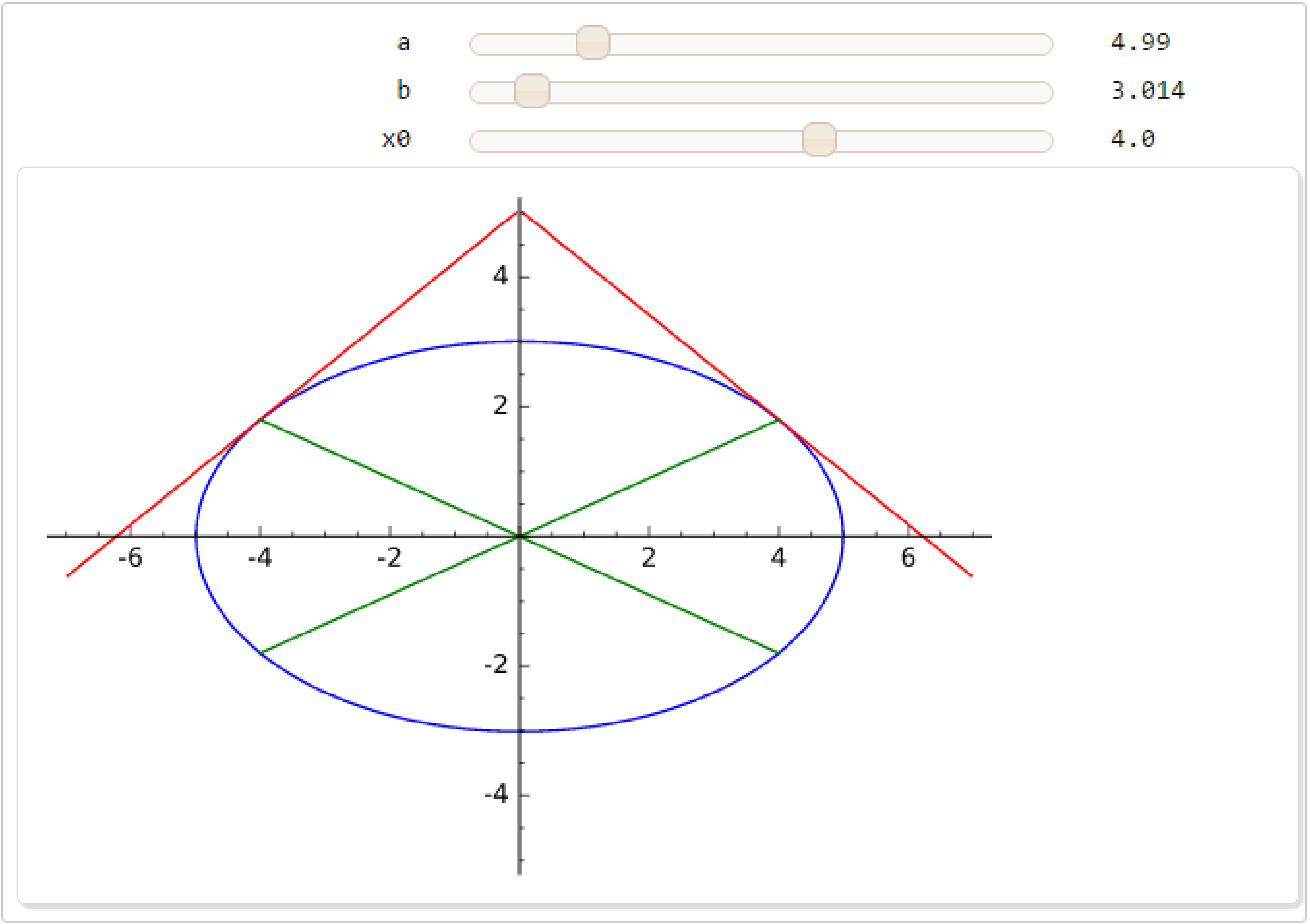
```
@interact
def my_plot(a=(5, (1, 20)), b=(3, (1, 20)), x0=(4, (-20,20))):
    if (x0>-a) and (x0<a):
        g1=implicit_plot(x^2/a^2 + y^2/b^2 == 1, (x, -a-2, a+2), (y, -b-2, b+2))
        g2=line(((x0, sqrt(b^2*(1-x0^2/a^2))), (-x0, -sqrt(b^2*(1-x0^2/a^2)))),
                                                        color='green')
        g3=line(((x0, -sqrt(b^2*(1-x0^2/a^2))), (-x0, +sqrt(b^2*(1-x0^2/a^2)))),
                                                        color='green')
        g4=implicit_plot(x0*x/a^2 + sqrt(b^2*(1-x0^2/a^2))*y/b^2 == 1, (x, -a-2,
                                          a+2), (y, -b-2, b+2), color='red')
        g5=implicit_plot(x0*x/a^2 - sqrt(b^2*(1-x0^2/a^2))*y/b^2 == 1, (x, -a-2,
                                          a+2), (y, -b-2, b+2), color='red')
        show(g1+g2+g3+g4+g5,axes=True,frame=False, aspect_ratio=1)
```


a
4.99
b
3.014
x0
4.0
4
2
-2
-4
-6
-4
-2
2
4
6

Άσκηση 11η: Να σχεδιάσετε την έλλειψη $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$ μαζί με τις εφαπτομένες της στα συμμετρικά σημεία της, ως προς τον άξονα Oy', καθώς και τις διαμέτρους που διέρχονται από αυτά για διάφορες τιμές των παραμέτρων a, b και x0.

Λύση:

```
@interact
def my_plot(a=(5, (1, 20)), b=(3, (1, 20)), x0=(4, (-20,20))):
    if (x0>-a) and (x0<a):
        g1=implicit_plot(x^2/a^2 + y^2/b^2 == 1, (x, -a-2, a+2), (y, -b-2, b+2))
        g2=line(((x0, sqrt(b^2*(1-x0^2/a^2))), (-x0, -sqrt(b^2*(1-x0^2/a^2)))),
                color='green')
        g3=line(((-x0, sqrt(b^2*(1-x0^2/a^2))), (x0, -sqrt(b^2*(1-x0^2/a^2)))),
                color='green')
        g4=implicit_plot(x0*x/a^2 + sqrt(b^2*(1-x0^2/a^2))*y/b^2 == 1, (x, -a-2,
                a+2), (y, -b-2, b+2), color='red')
        g5=implicit_plot(-x0*x/a^2 + sqrt(b^2*(1-x0^2/a^2))*y/b^2 == 1, (x, -a-2,
                a+2), (y, -b-2, b+2), color='red')
        show(g1+g2+g3+g4+g5,axes=True,frame=False, aspect_ratio=1)
```


a
4.99
b
3.014
x0
4.0
4
2
-2
-4
-6
-4
-2
2
4
6

Χρήστος Τσαγγάρης
ΕΔΙΠ Τμήματος Μαθηματικών

10-01-2017

Εφαπτομένη – Μετατόπιση - Στροφή

Εφαπτομένη 1/2

Η εφαπτομένη μίας καμπύλης (c) με δευτεροβάθμια εξίσωση:

$$A(x-x_0)^2+B(x-x_0)(y-y_0)+C(y-y_0)^2+2D(x-x_0)+2E(y-y_0)=0$$

στο σημείο $P(x_0, y_0)$ της (c) δίνεται από τον τύπο:

$$2D(x-x_0)+2E(y-y_0)=0$$

Άσκηση 1η: Να σχεδιάσετε την εφαπτομένη της έλλειψης:

$$x^2 + 4y^2 = 8$$

στο σημείο της P(2, -1).

Λύση:

$$x^2 + 4y^2 = 8 \Leftrightarrow (x - 2 + 2)^2 + 4(y + 1 - 1)^2 = 8 \Leftrightarrow$$

$$(x - 2)^2 + 4(x - 2) + 4 + 4(y + 1)^2 - 8(y + 1) + 4 = 8 \Leftrightarrow$$

$$(x - 2)^2 + 4(x - 2) + 4(y + 1)^2 - 8(y + 1) = 0$$

Άρα έχουμε την εφαπτομένη:

$$4(x - 2) - 8(y + 1) = 0 \Leftrightarrow (x - 2) - 2(y + 1) = 0 \Leftrightarrow$$

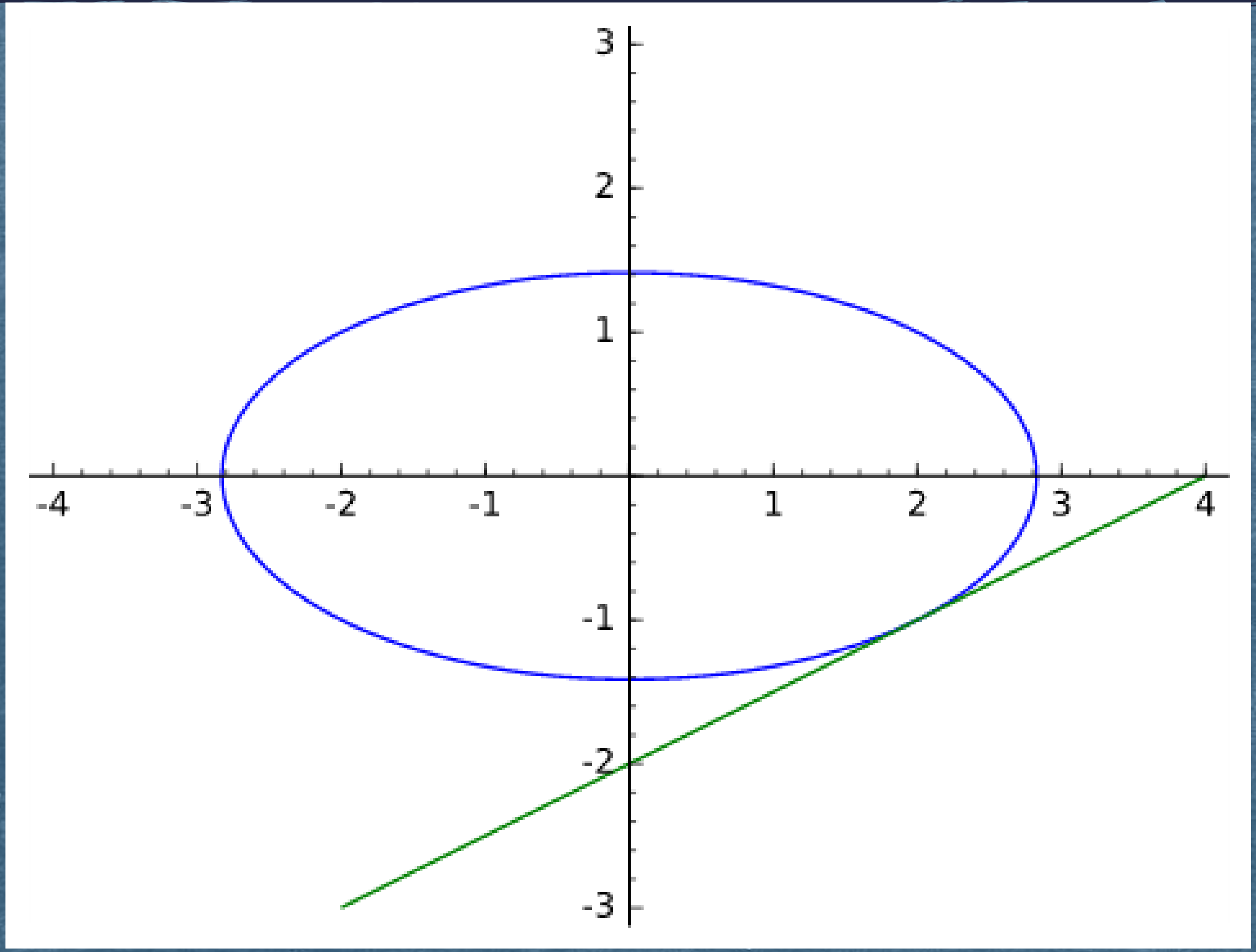
$$x - 2 - 2y - 2 = 0 \Leftrightarrow x - 2y - 4 = 0$$

Άσκηση 1η: Συνέχεια …

Ας σχεδιάσουμε την έλλειψη με την εφαπτομένη της στο σημείο P(2, -1).

```
y=var('y')

g1=implicit_plot(x^2 +4*y^2 == 8, (x, -4, 4), (y, -3, 3))

g2=implicit_plot(x-2*y-4==0, (x, -4, 4), (y, -3, 3), color='green')

show(g1+g2,axes=True,frame=False, aspect_ratio=1)
```

Άσκηση 1η: Συνέχεια …



Εφαπτομένη 2/2

Γενικεύοντας, η καμπύλη (c) με δευτεροβάθμια εξίσωση:

$$Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$$

έχει εφαπτομένη στο σημείο $P(x_0, y_0)$ την ευθεία με εξίσωση:

$$Ax_0x + B(x_0y + y_0x) + Cy_0y + D(x + x_0) + E(y + y_0) + F = 0$$

Άσκηση 2η: Να σχεδιάσετε τις εφαπτομένες της έλλειψης:

$$x^2 + 4y^2 = 8$$

στα διάφορα συμμετρικά σημεία της, ως προς τον άξονα xx'.

Λύση:

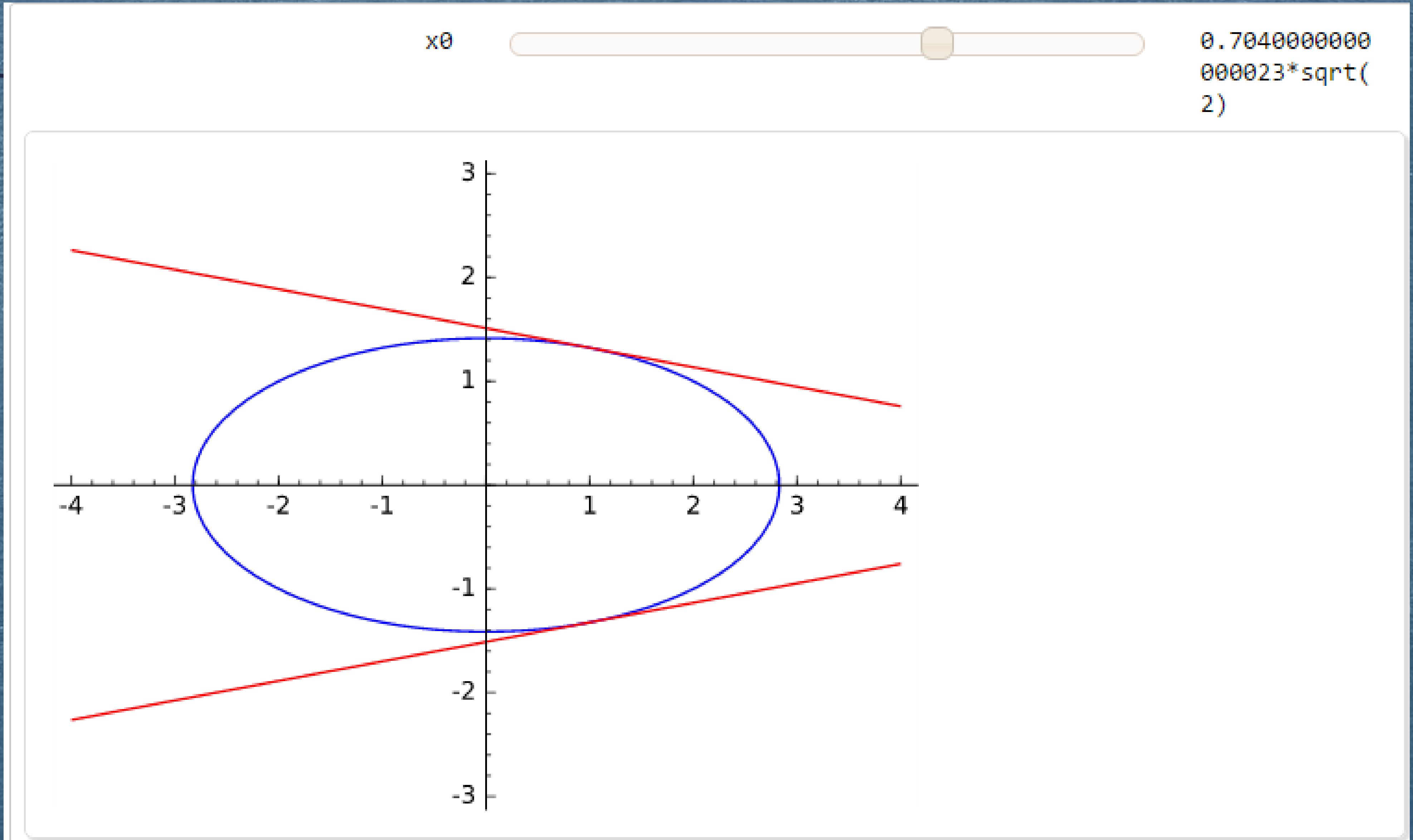
Εδώ προφανώς έχουμε: A=1, B=0, C=4, D=0, E=0 και F = -8. Οπότε η εφαπτομένη θα έχει την εξίσωση: $x_0x + 4y_0y - 8 = 0$.

Για ένα τυχαίο σημείο x_0 μεταξύ των κορυφών της έλλειψης $-\sqrt{8}$ και $\sqrt{8}$ έχουμε $x_0^2 + 4y_0^2 = 8 \Leftrightarrow y_0 = \pm\frac{\sqrt{8-x_0^2}}{2}$.

Άσκηση 2η: Συνέχεια ...

```
@interact

def my_plot(x0=(1, (-sqrt(8),sqrt(8)))):

    g1=implicit_plot(x^2 + 4*y^2 == 8, (x, -4, 4), (y, -3, 3))

    g2=implicit_plot(x0*x + 4*sqrt(8-x0^2)/2*y -8 == 0, (x, -4, 4), (y, -3, 3),
                                                                    color='red')

    g3=implicit_plot(x0*x - 4*sqrt(8-x0^2)/2*y -8 == 0, (x, -4, 4), (y, -3, 3),
                                                                    color='red')

    show(g1+g2+g3, axes=True, frame=False, aspect_ratio=1)
```

Άσκηση 2η: Συνέχεια …



Παράλληλη Μετατόπιση Ενός Συστήματος Αξόνων Oxy

Οι τύποι της παράλληλης μετατόπισης του Oxy στο O'x'y' είναι:

$$x = x_1 + x_0$$
$$y = y_1 + y_0$$

ή

$$x_1 = x - x_0$$
$$y_1 = y - y_0$$

όπου

(x, y) οι συντεταγμένες ως προς το σύστημα Oxy,

(x_1, y_1) οι συντεταγμένες ως προς το σύστημα Ox'y',

(x_0, y_0) οι συντεταγμένες του O' στο σύστημα O'x'y'.

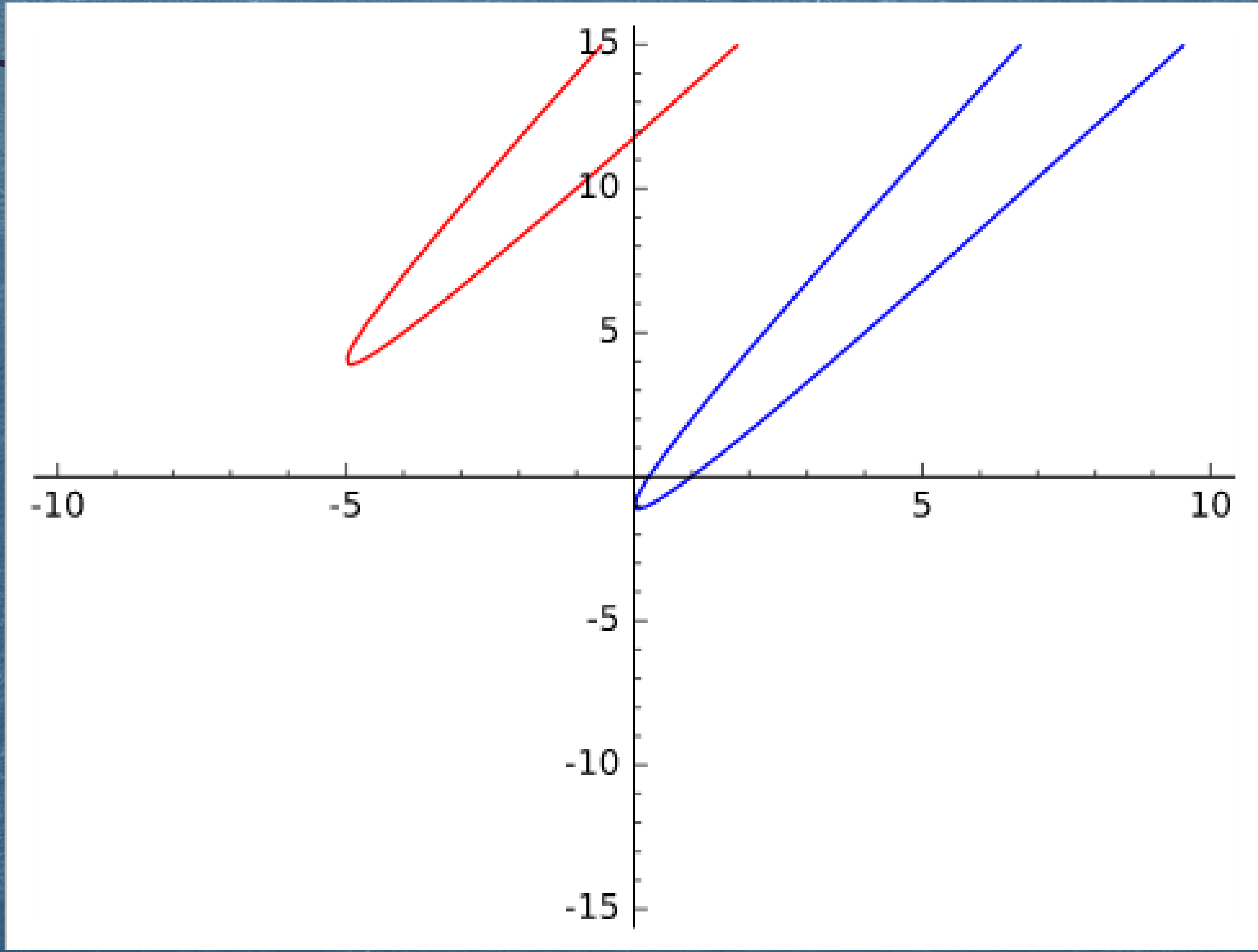
Οπότε για τη μετατόπιση στο νέο σύστημα O'x'y' στην εξίσωση που έχουμε αντικαθιστούμε το x με το x_1 δηλαδή με το $x-x_0$ και y με το y_1 δηλαδή με το $y-y_0$.

Άσκηση 3η: Αρχικά σχεδιάσετε την παραβολή $(y-2x+1)^2 = x$ και στη συνέχεια να τη μετατοπίσετε σε ένα νέο σύστημα αξόνων O'x'y' με O'(-5, 5).

Λύση:

```
g1=implicit_plot((y-2*x+1)^2 == x, (x, -10, 10), (y, -15, 15))

g2=implicit_plot(((y-5)-2*(x+5)+1)^2 == (x+5), (x, -10, 10), (y, -15, 15),
                                                        color="red")

show(g1+g2, axes=True, frame=False, aspect_ratio=0.5 )
```

Άσκηση 3η: Συνέχεια …

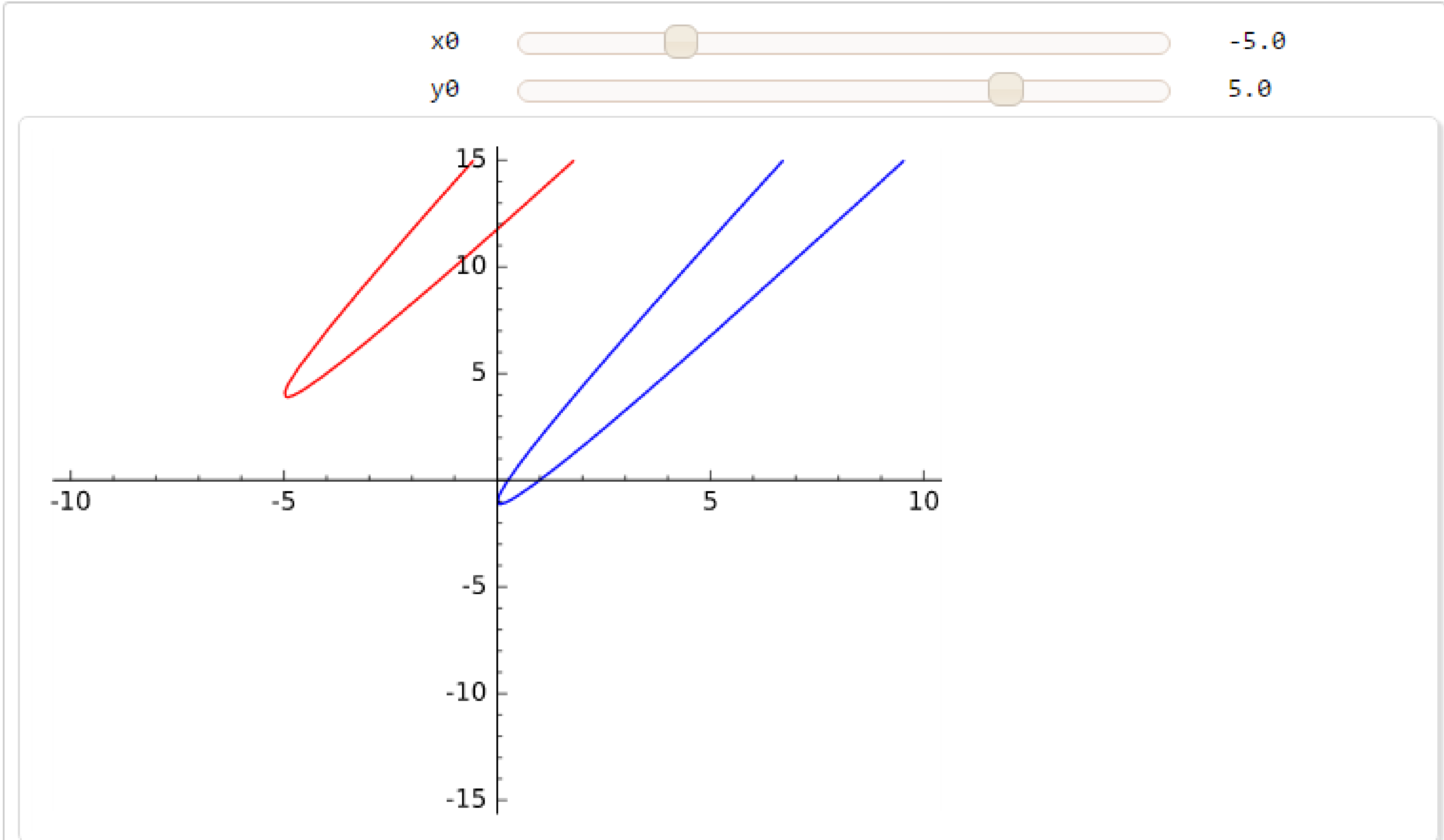


Άσκηση 4η: Αφού σχεδιάσετε τη την παραβολή $(y-2x+1)^2 = x$ να την μετατοπίσετε σε ένα νέο σύστημα αξόνων O’x’y’ με O’(x_0, y_0), για διάφορες τιμές των x_0, και y_0.

Λύση:

```
@interact

def my_plot(x0=(-5, (-10, 10)), y0=(5,(-10,10))):

    g1=implicit_plot((y-2*x+1)^2 == x, (x, -10, 10), (y, -15, 15))

    g2=implicit_plot(((y-y0)-2*(x-x0)+1)^2 == (x-x0), (x, -10, 10), (y, -15, 15),
                                                                    color="red")

show(g1+g2, axes=True, frame=False, aspect_ratio=0.5 )
```

Άσκηση 4η: Συνέχεια ...



Στροφή Ενός Συστήματος Αξόνων Oxy

Οι τύποι της στροφής ενός συστήματος Oxy κατά γωνία θ είναι:

$$x = x_1 * \cos\theta + y_1 * \sin\theta$$

$$y = x_1 * \sin\theta + y_1 * \cos\theta$$

ή

$$x_1 = x * \cos\theta + y * \sin\theta$$

$$y_1 = y * \cos\theta - x * \sin\theta$$

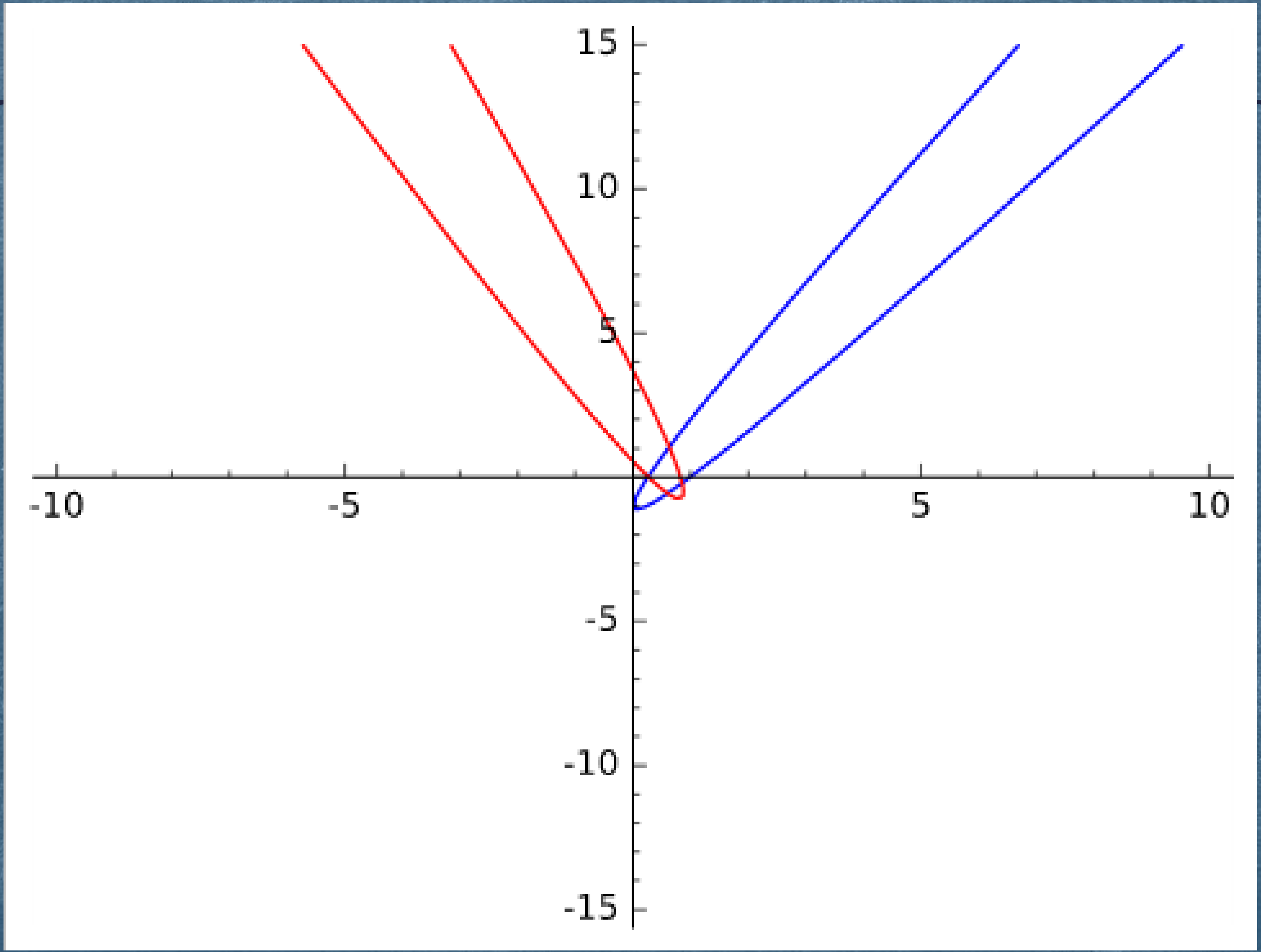
Οπότε για τη στροφή κατά γωνία θ, στην εξίσωση που έχουμε αντικαθιστούμε το x με το x_1 δηλαδή με το $x*\cos\theta + y*\sin\theta$ και y με το y_1 δηλαδή με το $y*\cos\theta - x*\sin\theta$.

Άσκηση 5η: Αρχικά σχεδιάσετε την παραβολή $(y\text{-}2x+1)^2 = x$ και στη συνέχεια να τη στρέψετε κατά γωνία θ=45°.

Λύση:

```
g1=implicit_plot((y-2*x+1)^2 == x, (x, -10, 10), (y, -15, 15))

g2=implicit_plot(((y*cos(pi/4)-x*sin(pi/4))-
                 2*(x*cos(pi/4)+y*sin(pi/4))+1)^2 == (x*cos(pi/4)+y*sin(pi/4)),
                 (x, -10, 10), (y, -15, 15),color="red")

show(g1+g2, axes=True, frame=False, aspect_ratio=0.5 )
```

Άσκηση 5η: Συνέχεια …

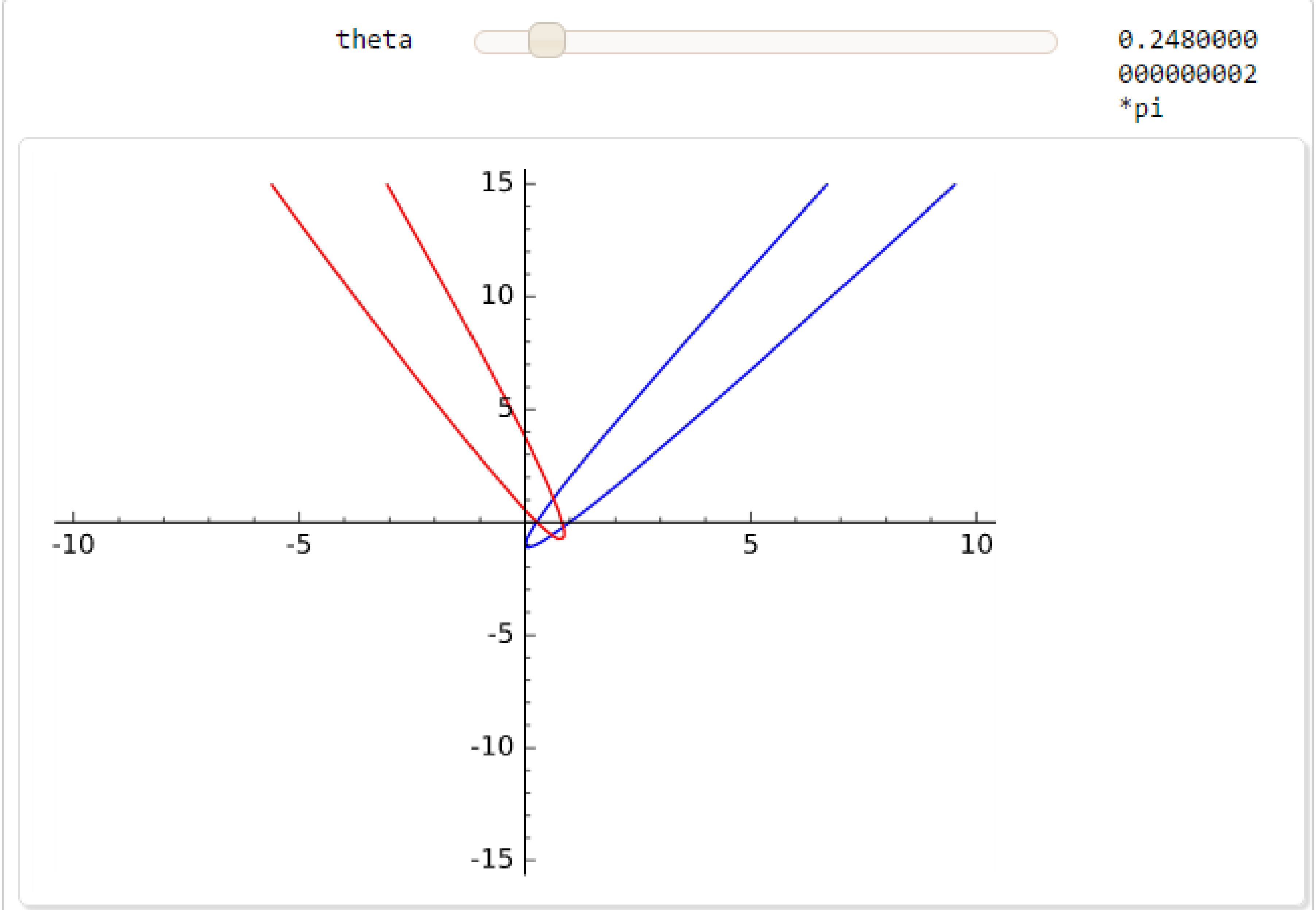


Άσκηση 6η: Αφού σχεδιάσετε τη την παραβολή $(y-2x+1)^2 = x$ να την στρέψετε κατά γωνία θ, για διάφορες τιμές της γωνίας θ.

Λύση:

```
theta=var('theta')

@interact

def my_plot(theta=(pi/4, (0, 2*pi))):

    g1=implicit_plot((y-2*x+1)^2 == x, (x, -10, 10), (y, -15, 15))

    g2=implicit_plot(((y*cos(theta)-x*sin(theta))
        -2*(x*cos(theta)+y*sin(theta))+1)^2 ==(x*cos(theta)+y*sin(theta)),
        (x, -10, 10), (y, -15, 15),color="red")

    show(g1+g2, axes=True, frame=False, aspect_ratio=0.5 )
```

Άσκηση 6η: Συνέχεια …



11-01-2017

Χρήστος Τσαγγάρης
ΕΔΙΠ Τμήματος Μαθηματικών

Αναλλοίωτες επιφάνειας β’ βαθμού

Εύρεση αναλλοίωτων επιφάνειας β’ βαθμού 1/2

Κάθε εξίσωση επιφάνειας β’ βαθμού έχει γενικό τύπο:

$$a_{11}*x^2 + a_{22}*y^2 + a_{33}*z^2 + 2a_{12}*xy + 2a_{13}*xz + 2a_{23}*yz + 2a_1*x + 2a_2*y + 2a_3*z + a_4 = 0$$

και οι αναλλοίωτες της ορίζονται ως εξής:

$J_1 = a_{11} + a_{22} + a_{33}$.

$$J_2 = \mathrm{Det}\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} + \mathrm{Det}\begin{pmatrix} a_{11} & a_{13} \\ a_{13} & a_{33} \end{pmatrix} + \mathrm{Det}\begin{pmatrix} a_{22} & a_{23} \\ a_{23} & a_{33} \end{pmatrix}$$

$$J_3 = \mathrm{Det}\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \quad \text{και} \quad J_4 = \mathrm{Det}\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_1 \\ a_{12} & a_{22} & a_{23} & a_2 \\ a_{13} & a_{23} & a_{33} & a_3 \\ a_1 & a_2 & a_3 & a_4 \end{pmatrix}$$

Ορισμός Πίνακα στο Sage

Στο Sage ένας πίνακας ορίζεται με τη χρήση της συνάρτησης ***matrix*** και δίνοντας τα στοιχεία του πίνακα κατά γραμμές μέσα σε αγκύλες. Για παράδειγμα αν θέλουμε να ορίσουμε τον πίνακας $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ γράφουμε:

matrix([[1,2], [3,4]])

και παίρνουμε ως αποτέλεσμα:

[1 2]
[3 4]

Ορίζουσα Πίνακα

Το Sage με τη συνάρτηση ***det*** υπολογίζει την ορίζουσα ενός πίνακα. Για παράδειγμα αν θέλουμε να υπολογίσουμε την ορίζουσα του πίνακα $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ γράφουμε:

det(matrix([[1,2], [3,4]]))

και παίρνουμε ως αποτέλεσμα:

-2.

Εύρεση αναλλοίωτων επιφάνειας β' βαθμού 2/2

Οπότε δοθέντος των συντελεστών a1, a2, a3, a4, a12, a13, a23, a11, a22 και a33 μπορούμε στο Sage να ορίσουμε τις αναλλοίωτες με τη χρήση των συναρτήσεων matrix και det ως εξής:

J1 = a11 + a22 + a33

J2 = det(matrix([[a11, a12], [a12, a22]])) +
det(matrix([[a11, a13], [a13, a33]])) +
det(matrix([[a22, a23], [a23, a33]]))

J3 = det (matrix([[a11, a12, a13], [a12, a22, a23], [a13, a23, a33]))

J4 = det (matrix([[a11, a12, a13, a1], [a12, a22, a23, a2], [a13, a23, a33, a3], [a1, a2, a3, a4]))

Άσκηση 1η: Να βρεθούν οι αναλλοίωτες της επιφάνειας:

$$x^2 + y^2 - z^2 + 2xz + x - 20 = 0.$$

Λύση:

```
a11=1; a22=1; a33=-1; a12=0; a13=1; a23=0; a1=1/2; a2=0; a3=0; a4=-20
J1 = a11 + a22 + a33
J2 = det( matrix( [ [a11, a12], [a12, a22] ] ) )
        +det( matrix( [ [a11, a13], [a13, a33] ] ) )
        +det( matrix( [ [a22, a23], [a23, a33] ] ) )
J3 = det( matrix ([[a11, a12, a13], [a12, a22, a23], [a13, a23, a33]] ) )
J4 = det( matrix ([[a11, a12, a13, a1], [a12, a22, a23, a2], [a13, a23, a33, a3],
                                                        [a1, a2, a3, a4]] ) )
print 'J1=', J1, ',  J2=', J2, ',  J3=', J3, ',  J4=', J4
```

```
J1= 1 , J2= -2 , J3= -2 , J4= 161/4
```

Άσκηση 2η: Να βρεθούν οι αναλλοίωτες της επιφάνειας:

$$5x^2 + 3y^2 + 4z^2 + 2xy + 8xz - 4x - 1 = 0.$$

Λύση:

```
a11=5; a22=3; a33=4; a12=1; a13=4; a23=0; a1=-2; a2=0; a3=0; a4=-1

J1 = a11 + a22 + a33

J2 = det( matrix( [ [a11, a12], [a12, a22] ] ) )
        +det( matrix( [ [a11, a13], [a13, a33] ] ) )
        +det( matrix( [ [a22, a23], [a23, a33] ] ) )

J3 = det( matrix ([[a11, a12, a13], [a12, a22, a23], [a13, a23, a33]] ) )

J4 = det( matrix ([[a11, a12, a13, a1], [a12, a22, a23, a2], [a13, a23, a33, a3],
                                                  [a1, a2, a3, a4]] ) )

print 'J1=', J1, ',  J2=', J2, ',  J3=', J3, ',  J4=', J4
```

J1= 12 , J2= 30 , J3= 8 , J4= -56

Είδος καμπύλης και τιμές αναλλοίωτων 1/4

Γνωρίζουμε ότι το είδος της καμπύλης εξαρτάται από τους συντελεστές του πολυωνύμου και επομένως οι τιμές των αναλλοίωτων που προκύπτουν προσδιορίζουν διαφορετικό είδος επιφάνειας. Συγκεκριμένα:

- Αν $J_3 \neq 0$ και $J_4 = 0$ τότε έχουμε «Πραγματικό Κώνο ή ένα σημείο».
- Αν $J_3 \neq 0$ και $J_4 > 0$ και ($J_2 > 0$ και $J_1 J_2 > 0$) τότε έχουμε «Κενό (φανταστικό ελλειψοειδές)».
- Αν $J_3 \neq 0$ και $J_4 > 0$ και ($J_2 \leq 0$ ή $J_1 J_2 \leq 0$) τότε έχουμε «Μονόχωνο υπερβολοειδές».

Είδος καμπύλης και τιμές αναλλοίωτων 2/4

- Αν $J_3 \neq 0$ και $J_4 < 0$ και ($J_2 > 0$ και $J_1 J_2 > 0$) τότε έχουμε «Πραγματικό ελλειψοειδές».
- Αν $J_3 \neq 0$ και $J_4 < 0$ και ($J_2 \leq 0$ ή $J_1 J_2 \leq 0$) τότε έχουμε «Δίχωνο υπερβολοειδές».
- Αν $J_3 = 0$ και $J_4 < 0$ τότε έχουμε «Παραβολοειδές ελλειπτικό».
- Αν $J_3 = 0$ και $J_4 > 0$ τότε έχουμε «Παραβολοειδές υπερβολικό».
- Αν $J_3 = 0$ και $J_4 = 0$ τότε έχουμε «Κύλινδρο ή δύο επίπεδα (πραγματικά ή φανταστικά».

Είδος καμπύλης και τιμές αναλλοίωτων 3/4

Ο παρακάτω κώδικας στο Sage υλοποιεί όλες τις προηγούμενες συνθήκες:

```
if J3!=0 and J4==0:
    print 'Πραγματικός Κώνος ή ένα σημείο'

elif J3 != 0 and J4 > 0 and (J2>0 and J1*J2>0):
    print 'Κενο (φανταστικό) ελλειψοειδές'

elif J3 != 0 and J4 > 0 and (J2<=0 or J1*J2<=0):
    print 'Μονόχωνο υπερβολοειδές'

elif J3 != 0 and J4 < 0 and (J2>0 and J1*J2>0):
    print 'Πραγματικό ελλειψοειδές'
```

Είδος καμπύλης και τιμές αναλλοίωτων 4/4

```
elif J3 != 0 and J4 < 0 and (J2<=0 or J1*J2<=0):
    print 'Δίχωνο υπερβολοειδές‘

elif J3 == 0 and J4 < 0 :
    print 'Παραβολοειδές ελλειπτικό'

elif J3 == 0 and J4 > 0 :
    print 'Παραβολοειδές υπερβολικό'

elif J3 == 0 and J4 == 0 :
    print 'Κύλινδρος ή δύο επίπεδα (πραγματικά ή φανταστικά)'

else:
    print 'Λάθος δεδομένα'
```

Άσκηση 3η: Να βρεθεί το είδος της καμπύλης:

$$5x^2 + 3y^2 + 4z^2 + 2xy + 8xz - 4x - 1 = 0.$$

Λύση:

Δεδομένα:

a11=5; a22=3; a33=4; a12=1; a13=4; a23=0; a1=-2; a2=0; a3=0; a4=-1

Αποτελέσματα:

J1= 12 , J2= 30 , J3= 8 , J4= -56

Πραγματικό ελλειψοειδές

Άσκηση 4η: Να βρεθεί το είδος της καμπύλης:

$$5x^2 + 5y^2 - 12z^2 + 2xy - 12 = 0.$$

Λύση:

Δεδομένα:

a11=5; a22=5; a33=-12; a12=1; a13=0; a23=0; a1=0; a2=0; a3=0; a4=-12

Αποτελέσματα:

J1= -2 , J2= -96 , J3= -288 , J4= 3456

Μονόχωνο υπερβολοειδές

Άσκηση 5η: Να σχεδιάσετε την καμπύλη:

$$y^2 - x^2 - z^2 - 1 = 0,$$

αφού πρώτα προσδιορίσετε το είδος της.

Λύση:

Δεδομένα:

a11=-1; a22=1; a33=-1; a12=0; a13=0; a23=0; a1=0; a2=0; a3=0; a4=-1

....

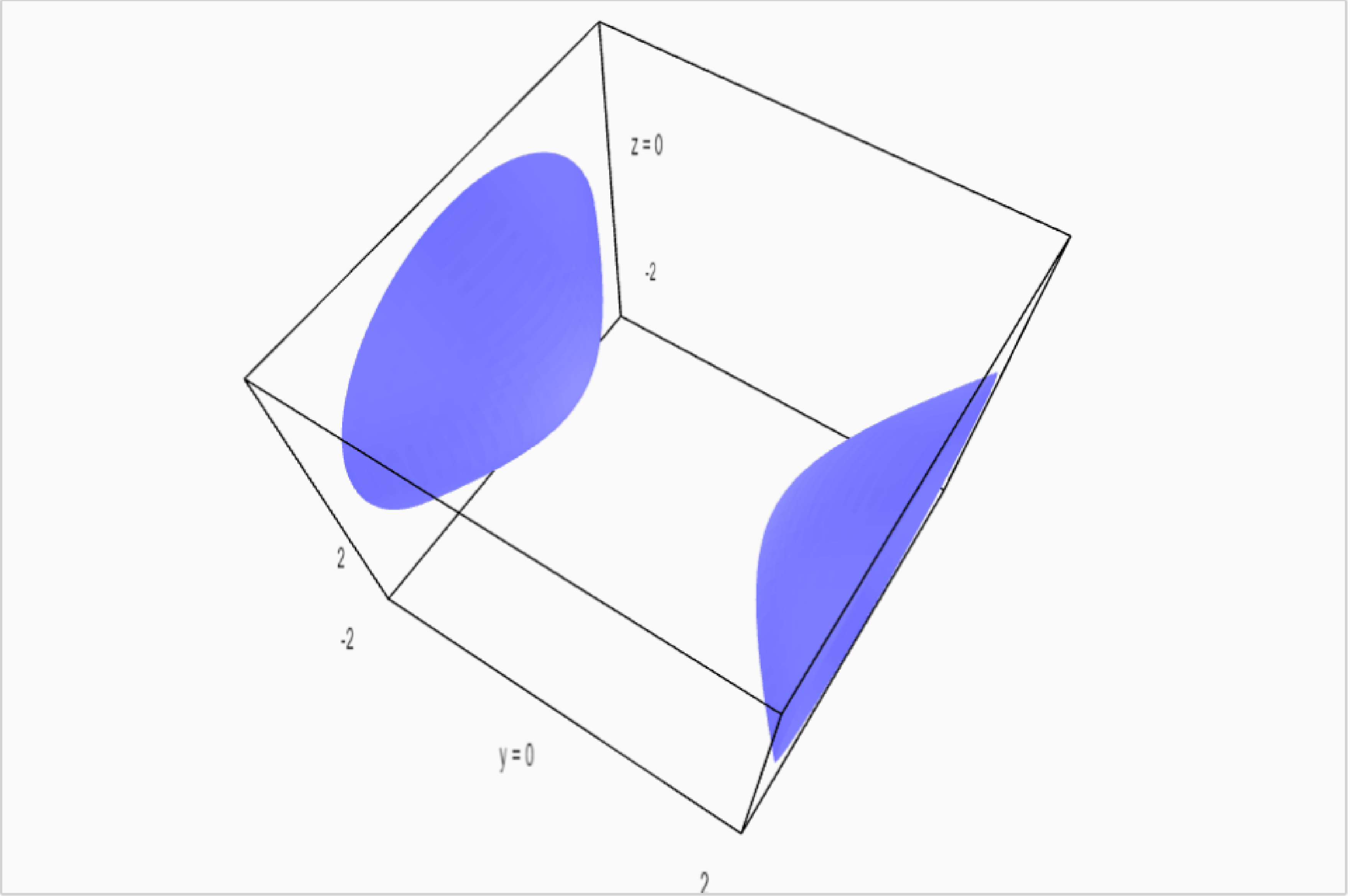
```
y,z=var('y z')
implicit_plot3d(y^2-x^2-z^2-1==0, (x, -2, 2), (y, -2, 2), (z, -2, 2))
```

Αποτελέσματα:

J1= -1 , J2= -1 , J3= 1 , J4= -1

Δίχωνο υπερβολοειδές

Άσκηση 5η: Συνέχεια …



Άσκηση 6η: Να σχεδιάσετε την καμπύλη:

$$-x^2 - y^2 + z^2 - 2xy + x - y - 20 = 0,$$

αφού πρώτα προσδιορίσετε το είδος της.

Λύση:

Δεδομένα:

a11=-1; a22=-1; a33=1; a12=-1; a13=0; a23=0; a1=1/2; a2=-1/2; a3=0; a4=-20

....

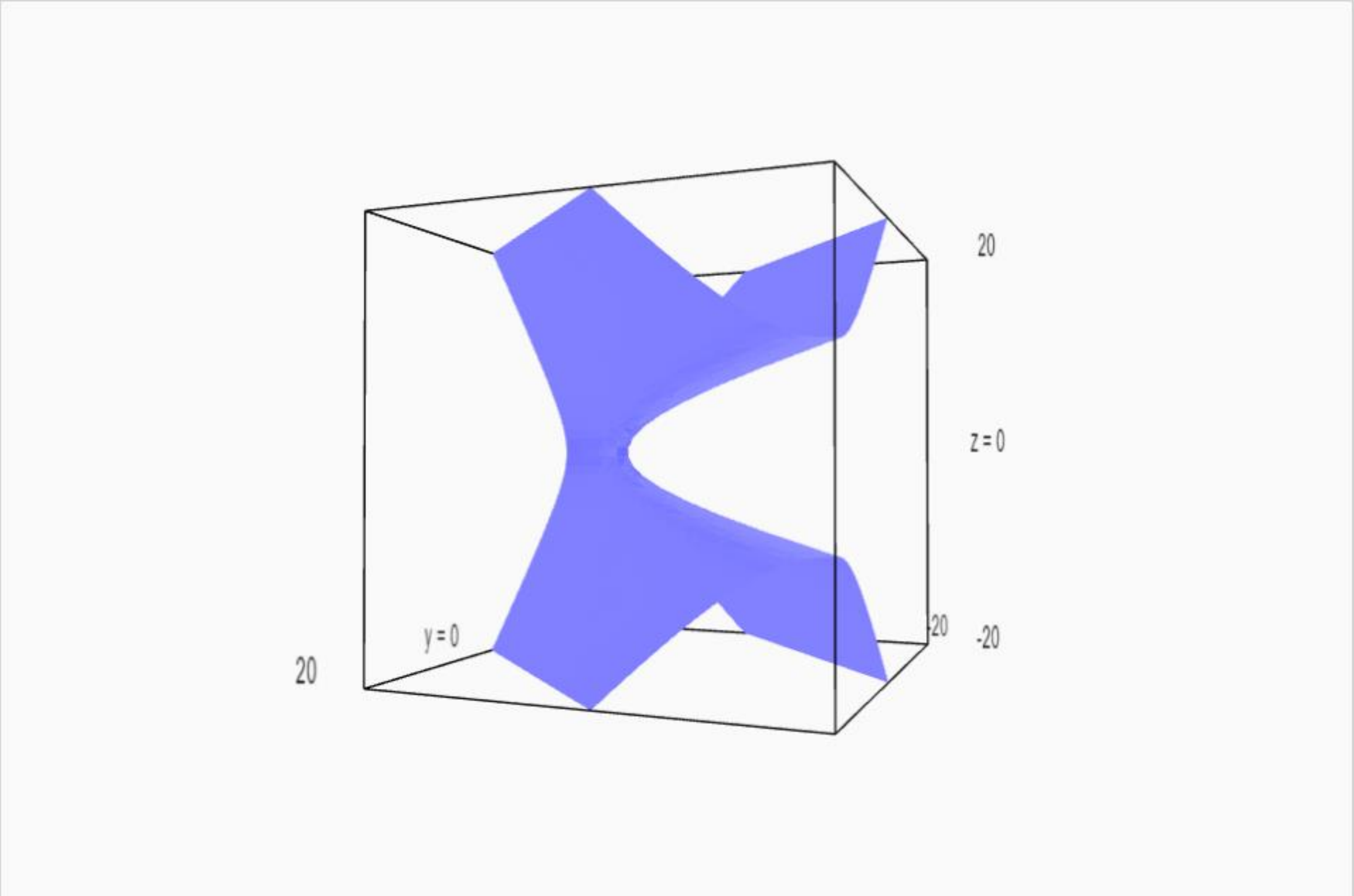
implicit_plot3d(-x^2-y^2+z^2+x-y-20==0, (x, -20, 20), (y, -20, 20), (z, -20, 20))

Αποτελέσματα:

J1= -1 , J2= -2 , J3= 0 , J4= 1

Παραβολοειδές υπερβολικό

Άσκηση 6η: Συνέχεια …



Άσκηση 7η: Να σχεδιάσετε την καμπύλη:

$$1.2x^2 + y^2 - z = 0,$$

αφού πρώτα προσδιορίσετε το είδος της.

Λύση:

Δεδομένα:

a11=1.2; a22=1; a33=0; a12=0; a13=0; a23=0; a1=0; a2=0; a3=-1/2; a4=0

....

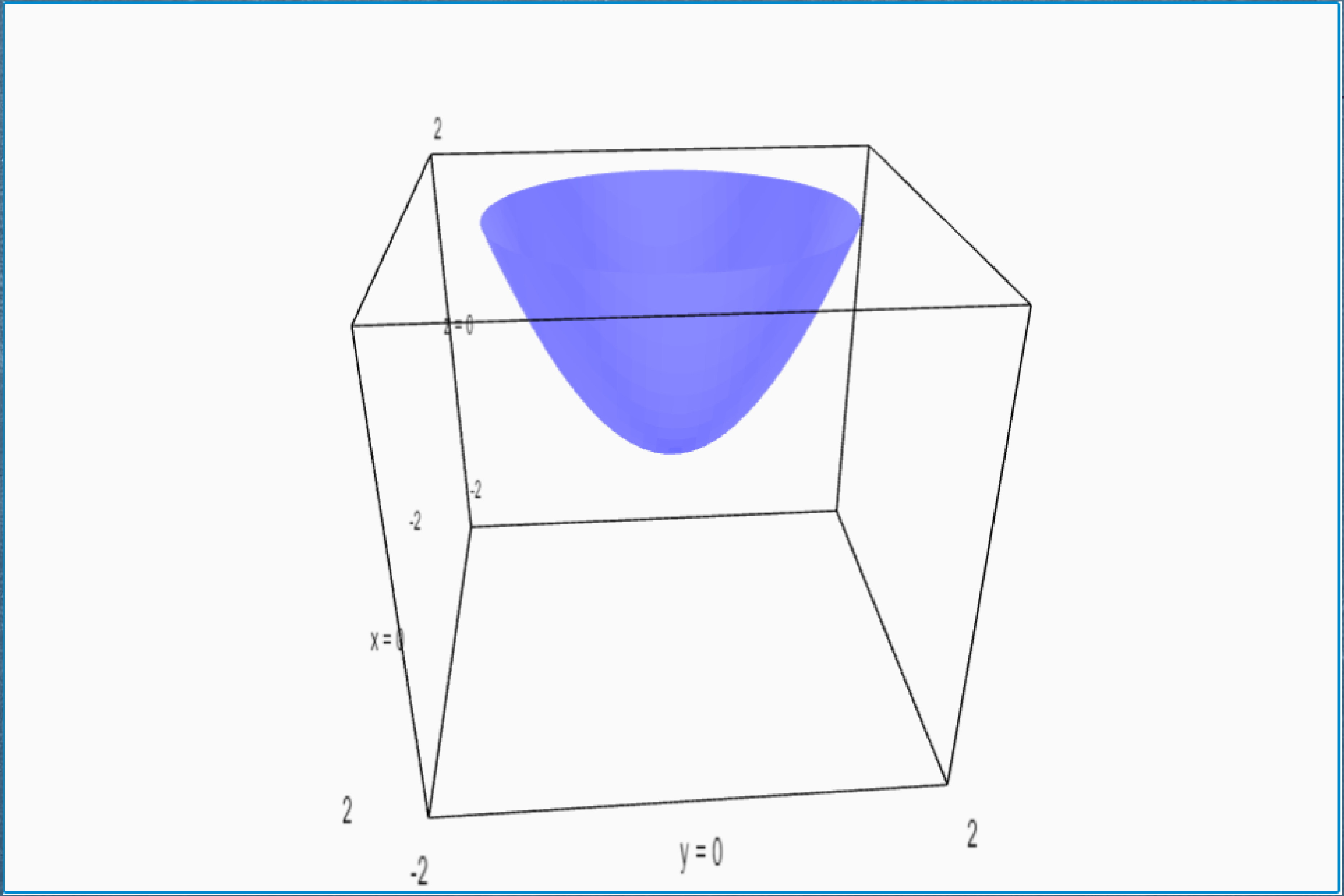
implicit_plot3d(1.2*x^2 + y^2 -z == 0, (x, -2, 2), (y, -2, 2), (z, -2, 2))

Αποτελέσματα:

J1= 2.20000000000000 , J2= 1.20000000000000 , J3= 0.000000000000000 , J4= -0.300000000000000

Παραβολοειδές ελλειπτικό

Άσκηση 7η: Συνέχεια …



Άσκηση 8η: Να σχεδιάσετε την καμπύλη:

$$5x^2 + 3y^2 - 2z^2 = 0,$$

αφού πρώτα προσδιορίσετε το είδος της.

Λύση:

Δεδομένα:

a11=5; a22=3; a33=-2; a12=0; a13=0; a23=0; a1=0; a2=0; a3=0; a4=0

....

implicit_plot3d(5*x^2 + 3*y^2 -2*z^2 == 0, (x, -2, 2), (y, -2, 2), (z, -2, 2))

Αποτελέσματα:

J1= 6 , J2= -1 , J3= -30 , J4= 0

Πραγματικός Κώνος ή ένα σημείο

Άσκηση 8η: Συνέχεια …