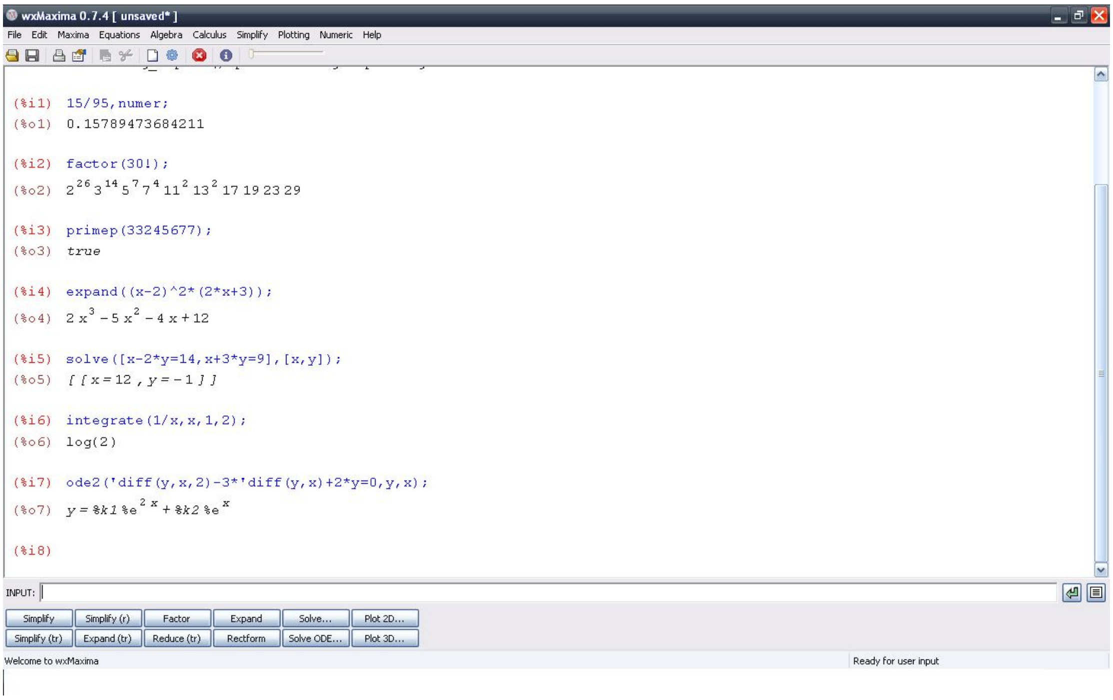
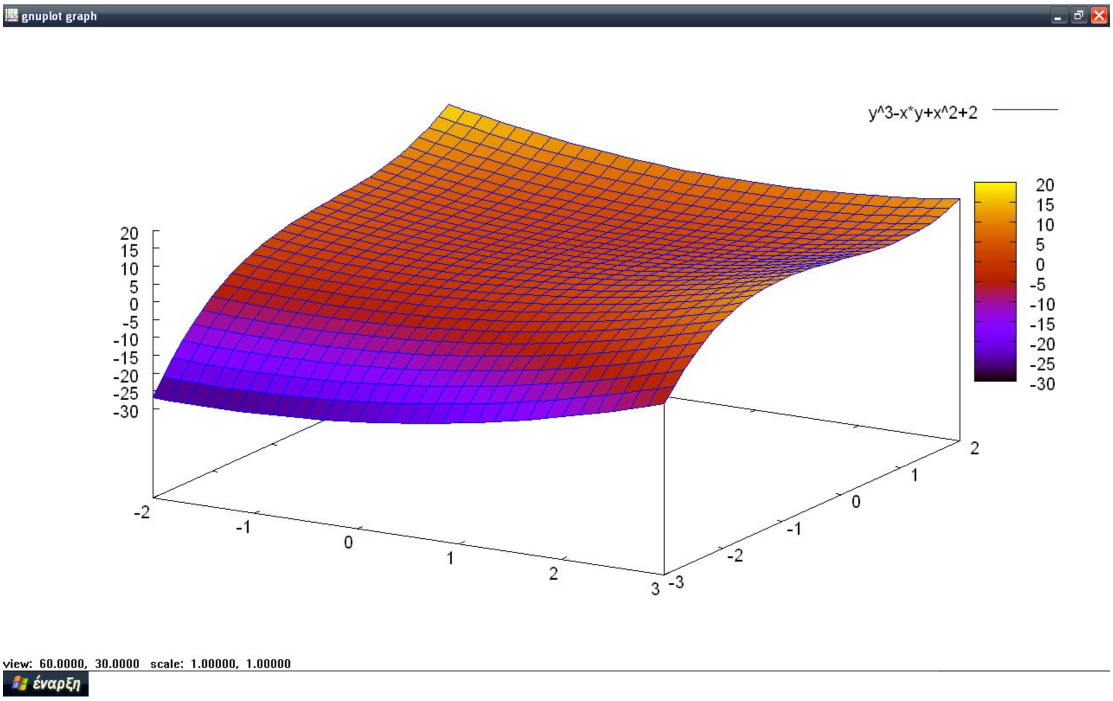
Maxima: οδηγίες χρήσης

Αλέξιος Ταμπαρόπουλος

alexis[dot]maxima[at]gmail[dot]com

Περιεχόμενα


wxMaxima 0.7.4 [unsaved*]
File Edit Maxima Equations Algebra Calculus Simplify Plotting Numeric Help
(%i1) 15/95,numer;
(%o1) 0.15789473684211
(%i2) factor(30!);
(%o2) 2 26 3 14 5 7 7 4 11 2 13 2 17 19 23 29
(%i3) primep(33245677);
(%o3) true
(%i4) expand((x-2)^2*(2*x+3));
(%o4) 2 x 3 − 5 x 2 − 4 x + 12
(%i5) solve([x-2*y=14,x+3*y=9],[x,y]);
(%o5) [[x = 12 , y = − 1]]
(%i6) integrate(1/x,x,1,2);
(%o6) log(2)
(%i7) ode2('diff(y,x,2)-3*'diff(y,x)+2*y=0,y,x);
(%o7) y = %k1 %e 2 x + %k2 %e x
(%i8)
INPUT:
Simplify
Simplify (r)
Factor
Expand
Solve...
Plot 2D...
Simplify (tr)
Expand (tr)
Reduce (tr)
Rectform
Solve ODE...
Plot 3D...
Welcome to wxMaxima
Ready for user input


gnuplot graph
y^3-x*y+x^2+2
20
15
10
5
0
-5
-10
-15
-20
-25
-30
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
view: 60.0000, 30.0000 scale: 1.00000, 1.00000
έναρξη

Εισαγωγή

Το Maxima είναι ένα υπολογιστικό μαθηματικό σύστημα (computer algebra system), γραμμένο στην γλώσσα Lisp. Το Maxima παρήχθη από το σύστημα Macsyma, το οποίο αναπτύχθηκε από το Αμερικανικό Πανεπιστήμιο MIT από το 1968 ως το 1982 ως τμήμα του σχεδίου Project MAC. Το MIT έδωσε ένα αντίγραφο του πηγαίου κώδικα (source code) στο Τμήμα Ενέργειας (Department of Energy) το 1982. Η εκδοχή αυτή είναι γνωστή ως DOE Macsyma. Ένα αντίγραφο του DOE Macsyma έγινε αντικείμενο εργασίας του William F. Schelter, καθηγητή του Πανεπιστημίου του Texas, από το 1982 έως το θάνατό του το 2001. Το 1998 ο Schelter εξασφάλισε την άδεια από το Τμήμα Ενέργειας να δημοσιοποιήσει τον πηγαίο κώδικα του DOE Macsyma υπό την άδεια GNU Public License, και το 2000 ξεκίνησε το σχέδιο Maxima (Maxima project) στο SourceForge για την ανάπτυξη του DOE Macsyma, το καλούμενο τώρα Maxima. Το Maxima διατίθεται δωρεάν για Windows, Linux και Mac.

Το Maxima υποστηρίζει μαθηματικούς υπολογισμούς όπως: παραγώγιση, ολοκλήρωση, σειρές Taylor, μετασχηματισμούς Laplace, συνήθεις διαφορικές εξισώσεις, συστήματα γραμμικών εξισώσεων, πολυώνυμα, διανύσματα, μητρώα, τανυστές κ.λπ.

Η επίσημη ιστοσελίδα του προγράμματος βρίσκεται στην ηλεκτρονική διεύθυνση: http://maxima.sourceforge.net/

Ακολουθεί ένας οδηγός χρήσης του προγράμματος στα ελληνικά. Από τις εκατοντάδες εντολές -και αντίστοιχες δυνατότητες- παρουσιάζονται ορισμένες βασικές λειτουργίες, μέσα από παραδείγματα. Το κείμενο θα ανανεώνεται κάθε 3 μήνες. Η παρούσα έκδοση είναι η 1.1 (Ιανουάριος 2008).

Αριθμητικές πράξεις

Το Maxima μπορεί να εκτελέσει απλές αριθμητικές πράξεις, αν και σπάνια θα το χρησιμοποιήσουμε για αυτόν τον σκοπό. Ο στόχος όμως είναι η εξοικείωση με το πρόγραμμα και τις εντολές.

Όταν εκκινεί το πρόγραμμα, εμφανίζεται η συμβολοσειρά <%i1>. Το γράμμα i αντιστοιχεί στη λέξη input (είσοδος/στοιχείο εισόδου), ο αριθμός 1 στη διάταξη, αφού είναι το πρώτο στοιχείο. Μπορούμε να δώσουμε μία ή περισσότερες εντολές, τις οποίες χωρίζουμε με το (ελληνικό) ερωτηματικό <;>. Οι πράξεις συμβολίζονται ως εξής (χωρίς το <>):

<+> η πρόσθεση

<-> η αφαίρεση

<*> ο πολλαπλασιασμός

</> η διαίρεση

<^> ή <**> η ύψωση σε δύναμη

Αν δώσουμε

(11-7)*(6/2+5)^2;

δηλαδή $(11-7)\cdot\left(\frac{6}{2}+5\right)^2$

και πιέσουμε <Enter>, το πρόγραμμα θα επιστρέψει τη γραμμή:

(%o1) 256

δηλαδή το o (output=αποτέλεσμα) είναι 256, ενώ πιο πάνω εμφανίζει τη γραμμή:

(%i1) (11-7)*(6/2+5)^2;

η οποία κρατάει την πληροφορία εισόδου, στην οποία μας επιτρέπει να αναφερθούμε ξανά στα επόμενα, αν θέλουμε. Χρησιμοποιώντας το σύμβολο <%> το πρόγραμμα μας επιτρέπει να αναφερθούμε στην αμέσως προηγούμενη γραμμή:

(%i2) % - 10;

Δίνουμε έτσι την εντολή στο πρόγραμμα να αφαιρέσει το 10 από το τελευταίο αποτέλεσμα που είχε δώσει. Επομένως, θα μας επιστρέψει τη γραμμή:

(%o2) 246

Αν γράψουμε:

(%i3) %o1 / 8;

δίνουμε την εντολή στο πρόγραμμα να αναφερθεί στο αποτέλεσμα %o1 και να το διαιρέσει δια 8:

(%o3) 32

Το πρόγραμμα επιστρέφει το αποτέλεσμα μιας διαίρεσης ως ανάγωγο κλάσμα:

(%i4) 15/95;

(%o4) $\frac{3}{19}$

Γενικότερα, το πρόγραμμα εκτελεί τις ζητούμενες πράξεις κάνοντας ακριβείς υπολογισμούς, χειριζόμενο ακέραιους, κλάσματα και σύμβολα (όπως τα ριζικά). Αν θέλουμε το αποτέλεσμα σε δεκαδική μορφή, χρησιμοποιούμε την εντολή float, οπότε αυτό εξάγεται προσεγγιστικά, αν ο αριθμός των δεκαδικών ψηφίων του υπερβαίνει την προκαθορισμένη ακρίβεια του προγράμματος.

(%i5) float(15/95);

(%o5) 0.15789473684211

Μπορούμε να δηλώσουμε ότι θέλουμε το αποτέλεσμα σε δεκαδικό εναλλακτικά και ως εξής:

(%i6) 15/95.0;

(%o6) 0.15789473684211

(%i7) 15.0/95;

(%o7) 0.15789473684211

(%i8) 15/95,numer;

(%o8) 0.15789473684211

Το Maxima είναι σε θέση να χειριστεί πολύ μεγάλους αριθμούς. Για να υπολογίσουμε το 33 παραγοντικό:

(%i9) 33!;

(%o9) 868331761881188649551819440128000000

Με την εντολή float μπορούμε να μετατρέψουμε τον αριθμό σε τυποποιημένη εκθετική μορφή:

(%i10) float(%);

(%o10) $8.6833176188118859 \cdot 10^{+36}$

Αν το αποτέλεσμα μιας δύναμης είναι άρρητος, το πρόγραμμα δίνει το αποτέλεσμα ως δόθηκε.

(%i11) (1 + sqrt(2))^5;

(%o11) $\left(\sqrt{2}+1\right)^5$

Αν όμως θέλουμε να αναπτύξουμε τη δύναμη, χρησιμοποιήσουμε την εντολή expand.

(%i12) expand(%);

(%o12) $29\sqrt{2}+41$

Και για προσεγγιστικό υπολογισμό, όπως είδαμε και στη διαίρεση:

(%i13) float(%);

(%o13) 82.01219330881976

ή

(%i14) %i13,numer;

(%o14) 82.01219330881975

Βασικές σταθερές και συναρτήσεις

Μπορούμε να αναφερθούμε σε ορισμένες μαθηματικές σταθερές ως εξής:

%e - ο αριθμός e του Euler (2.71828…)

%pi - ο αριθμός π (3.14159…)

%phi - ο χρυσός αριθμός $\phi = \frac{\sqrt{5}+1}{2} = 1.61803...$

%i - η φανταστική μονάδα $i = \sqrt{-1}$

inf - το (+) άπειρο

minf - το (-) άπειρο

Ορισμένες βασικές συναρτήσεις είναι οι εξής: sin (ημίτονο), cos (συνημίτονο), tan (εφαπτομένη), cot (συνεφαπτομένη), sec (τέμνουσα), csc (συντέμνουσα), atan το τόξο εφαπτομένης, sqrt (τετραγωνική ρίζα), log (φυσικός λογάριθμος), exp (εκθετική συνάρτηση). Οι συναρτήσεις πρέπει να διαχωριστούν εννοιολογικά από τις εντολές και τους τελεστές, όπως η εντολή float που είδαμε ήδη και ο τελεστής limit που θα δούμε παρακάτω, αντίστοιχα. Ας δούμε τις συναρτήσεις στην πράξη:

```
(%i1) (sin(%pi/2)+ cos(%pi/3))* sqrt(81);
(%o1) 27/2
(%i2) exp(%pi*%i);
(%o2) -1
```

Μπορούμε να καθορίσουμε κάποιες παραμέτρους, τις οποίες σκοπεύουμε να χρησιμοποιήσουμε, με το σύμβολο <:>

```
(%i3) a:3;b:5;
(%o3) 3
(%o4) 5
```

Αυτό σημαίνει ότι στα επόμενα, όταν το πρόγραμμα συναντά τα γράμματα a και b, θα τα αντικαθιστά με τις τιμές 3 και 5 αντίστοιχα.

(%i5) sqrt(b^2-a^2);

(%o5) 4

Με ανάλογο τρόπο, με το σύμβολο <:=> ορίζουμε και τις συναρτήσεις που θέλουμε:

(%i6) f(x):=x^2-x+1;

(%o6) $f(x):=x^2-x+1$

(%i7) f(b)-f(a);

(%o7) 14

Σημειώνεται ότι το πρόγραμμα δεν διαθέτει ως συνάρτηση τον λογάριθμο με βάση το 10, αλλά μπορούμε να ορίσουμε τη συνάρτηση εμείς και να τη χρησιμοποιούμε στα επόμενα:

(%i8) log10(x):= log(x)/log(10);

(%o8) $\mathrm{log10}(x):=\dfrac{\log(x)}{\log(10)}$

Για να υπολογίσουμε αθροίσματα μιας ακολουθίας χρησιμοποιούμε την εντολή sum. Ας υποθέσουμε ότι ζητούμε το άθροισμα των όρων 1/n όπου το n λαμβάνει τις ακέραιες τιμές από 1 ως 100:

(%i9) sum(1/n,n,1,100);

(%o9) $\dfrac{14466636279520351160221518043104131447711}{2788815009188499086581352357412492142272}$

και για δεκαδική μορφή:

(%i10) %,numer;

(%o10) 5.18737751763962

Τα γινόμενα υπολογίζονται με την εντολή product:

(%i11) product(1+1/n^2,n,1,100),numer;

(%o11) 3.639682294531309

Θεωρία αριθμών

Η εντολή factor αναλύει έναν αριθμό σε γινόμενο πρώτων παραγόντων:

(%i1) factor(30!);

(%o1) $2^{26}\cdot 3^{14}\cdot 5^{7}\cdot 7^{4}\cdot 11^{2}\cdot 13^{2}\cdot 17\cdot 19\cdot 23\cdot 29$

Αν δώσουμε ένα ζεύγος αριθμών, η εντολές quotient και remainder μας επιστρέφουν το πηλίκο και το υπόλοιπο της διαίρεσης του πρώτου προς τον δεύτερο, αντίστοιχα:

(%i2) quotient(138,17);

(%o2) 8

(%i3) remainder(138,17);

(%o3) 2

Οι εντολές gcd (greatest common divisor) και lcm (least common multiple) δίνουν τον μέγιστο κοινό διαιρέτη και το ελάχιστο κοινό πολλαπλάσιο 2 αριθμών. Προκειμένου να χρησιμοποιήσουμε την εντολή lcm πρέπει να φορτώσουμε πρώτα ένα σύνολο εντολών, δίνοντας την εντολή load("functs"):

(%i4) gcd(18,30);

(%o4) 6

(%i5) load("functs");

(%i6) lcm(18,30);

(%o6) 90

Η εντολή gcd δέχεται ως όρισμα μόνο ένα ζεύγος αριθμών. Αν θέλουμε να βρούμε τον μκδ τριών ή περισσότερων αριθμών, μπορούμε να θυμηθούμε την ιδιότητα (a,b,c)=((a,b),c). Για τον μκδ(12,15,18):

(%i7) gcd(gcd(12,15),18);

(%o7) 3

Η εντολή primep ελέγχει αν ο δοσμένος αριθμός είναι ή όχι πρώτος και επιστρέφει την αληθοτιμή (true ή false):

(%i8) primep(33245677);

(%o8) true

Η εντολή divsum υπολογίζει το άθροισμα των διαιρετών ενός αριθμού:

(%i9) divsum(10);

(%o9) 18

Για να υπολογίσουμε το άθροισμα των k-δυνάμεων των διαιρετών ενός φυσικού n ($\sum_{d/n} d^k$), δίνουμε στην εντολή divsum ως όρισμα ένα ζεύγος αριθμών, όπου ο πρώτος δηλώνει τον αριθμό n και ο δεύτερος την επιθυμητή δύναμη k:

(%i10) divsum(10,3);

(%o10) 1134

Η εντολή binomial δίνει τον διωνυμικό συντελεστή $\binom{n}{m} = \frac{n!}{m!(n-m)!}$

(%i11) binomial(10,4);

(%o11) 210

Σύνολα

Το σύνολο δηλώνεται με την εντολή set, ή με άγκιστρα, αναγράφοντας τα στοιχεία του συνόλου. Αν δεν δώσουμε κανένα στοιχείο, αναφερόμαστε στο κενό σύνολο.

(%i1) a:set(1,2,3,4,5);

(%o1) {1,2,3,4,5}

(%i2) b:set(3,4,5,6,7);

(%o2) {3,4,5,6,7}

Για να βρούμε την ένωση (union) ή την τομή (intersection) των a και b:

(%i3) c:union(a,b);

(%o3) {1,2,3,4,5,6,7}

(%i4) d:intersection(a,b);

(%o4) {3,4,5}

Με τις εντολές powerset και cartesian_product βρίσκουμε το δυναμοσύνολο ενός συνόλου και το καρτεσιανό γινόμενο δύο συνόλων αντίστοιχα:

(%i5) powerset(d);

(%o5) {{},{3},{3,4},{3,4,5},{3,5},{4},{4,5},{5}}

(%i6) cartesian_product(a,d);

(%o6)

{[1,3],[1,4],[1,5],[2,3],[2,4],[2,5],[3,3],[3,4],[3,5],[4,3],[4,4],[4,5],[5,3],[5,4],[5,5]}

Η διαφορά 2 συνόλων A-B ή A\B δηλώνεται με την εντολή setdifference:

(%i7) setdifference(a,b);

(%o7) {1,2}

Η εντολή cardinality δίνει τον πληθάριθμο ενός συνόλου, το πλήθος των στοιχείων του:

(%i8) cardinality(c);

(%o8) 7

Πολυώνυμα και ρητές παραστάσεις

Ο τελεστής factor αναλύει ένα πολυώνυμο σε γινόμενο πρώτων παραγόντων. Η ανάλυση γίνεται στο Z[X]. Αυτό σημαίνει ότι μπορούμε να κάνουμε ανάλυση της μορφής: $\left(x^2+x-6\right)=\left(x+3\right)\left(x-2\right)$

αλλά όχι $\left(x^2+1\right)=\left(x-i\right)\left(x+i\right)$ ή $\left(x^2-2\right)=\left(x-\sqrt{2}\right)\left(x+\sqrt{2}\right)$

Για να παραγοντοποιήσουμε την ακόλουθη παράσταση:

$3x^9-11x^8+22x^7-38x^6+48x^5-48x^4+42x^3-26x^2+13x-5$

(%i1) factor(3*x^9-11*x^8+22*x^7-38*x^6+48*x^5-48*x^4+42*x^3-26*x^2+13*x-5);

(%o1) $\left(x-1\right)^2\left(3x-5\right)\left(x^2+1\right)^3$

Ο τελεστής expand λειτουργεί αντίστροφα. Αναπτύσσει δυνάμεις και γινόμενα πολυωνύμων:

(%i2) expand((x-2)^2*(2*x+3));

(%o2) $2x^3-5x^2-4x+12$

Μπορούμε να αντικαταστήσουμε κάποιες μεταβλητές με άλλες, ή με αριθμητικές τιμές, ως εξής:

(%i3) expand((x+y+1)^2);

(%o3) $y^2+2xy+2y+x^2+2x+1$

(%i4) %,x=y,y=z,z=1;

(%o4) $z^2+2yz+2z+y^2+2y+1$

Με τη χρήση του τελεστή solve λύνουμε αλγεβρικές εξισώσεις. Στο όρισμα πρέπει να δώσουμε την εξίσωση (αν παραλείψουμε το =0, το πρόγραμμα θα το θεωρήσει δεδομένο) και να καθορίσουμε ποιος είναι ο άγνωστος της εξίσωσης.

(%i5) solve(x^2-4=0,x);

(%o5) [x=-2,x=2]

Το πρόγραμμα επιστρέφει τη λύση μιας εξίσωσης, εφόσον η τελευταία είναι επιλύσιμη με ριζικά. Είναι γνωστό ότι οι πολυωνυμικές εξισώσεις βαθμού 5 ή μεγαλύτερου δεν έχουν γενικά τέτοια “κλειστή” λύση, οπότε το πρόγραμμα δεν θα επιστρέψει αποτέλεσμα. Το πρόγραμμα όμως θα εξετάσει από μόνο του την ύπαρξη ανάλυσης του πολυωνύμου σε γινόμενο παραγόντων (σύμφωνα με όσα συζητήθηκαν για την εντολή factor) και θα λύσει χωριστά τις εξισώσεις που προκύπτουν, δίνοντας τις ακριβείς τιμές των ριζών, όπου υπάρχουν. Αν, για παράδειγμα, ένα πολυώνυμο βαθμού 7 αναλύεται σε γινόμενο δύο αναγώγων πολυωνύμων βαθμού 4 και 3 αντίστοιχα, θα πάρουμε την πλήρη λύση. Αν όμως αναλύεται σε γινόμενο τριών αναγώγων πολυωνύμων, ενός βαθμού 5 και δύο πρωτοβάθμιων, θα πάρουμε μόνο 2 ρίζες. Ένα τέτοιο παράδειγμα είναι η εξίσωση $x^7-2x^6-4x^5+13x^4-12x^3+5x^2-3x+2=0$:

(%i6) solve(x^7-2*x^6-4*x^5+13*x^4-12*x^3+5*x^2-3*x+2);

(%o6) $[x=1, x=2, 0=x^5+x^4-3x^3+2x^2+1]$

Το πρόγραμμα μπορεί να λύσει και παραμετρικές εξισώσεις. Για την εξίσωση $x^2-ax+1=0$:

(%i7) solve(x^2-a*x+1,x);

(%o7) $[x=-\frac{\sqrt{a^2-4}-a}{2}, x=\frac{\sqrt{a^2-4}+a}{2}]$

Η εντολή nroots υπολογίζει το πλήθος των πραγματικών ριζών μιας πολυωνυμικής εξίσωσης σε ένα δοσμένο διάστημα της μορφής (,]. Ας υπολογίσουμε το πλήθος των πραγματικών λύσεων της εξίσωσης $x^7-3x^4+1=0$ στο διάστημα (-5,3]:

(%i8) p:x^7-3*x^4+1;

(%o8) x^7-3x^4+1

(%i9) nroots(p,-5,3);

(%o9) 3

Το Maxima λύνει τις εξισώσεις στο C, δίνοντας πραγματικές και μιγαδικές ρίζες:

(%i10) solve(x^2+2*x+5,x);

(%o10) [x=-2*%i-1,x=2*%i-1]

Για να λύσουμε ένα σύστημα γραμμικών εξισώσεων, δίνουμε ένα όρισμα που αποτελείται από δύο ζεύγη αγκυλών. Στο πρώτο τοποθετούμε τις εξισώσεις, χωρισμένες με κόμμα και στο δεύτερο τους αγνώστους, επίσης χωρισμένους με κόμμα. Αν οι εξισώσεις έχουν μόνο πρώτο μέλος, το δεύτερο μέλος τίθεται αυτόματα =0. Για το σύστημα:

$$\begin{cases} x-2y=14 \\ x+3y=9 \end{cases}$$

θα γράψουμε:

(%i11) solve([x-2*y=14,x+3*y=9],[x,y]);

(%o11) [[x=12,y=-1]]

Για το παραμετρικό σύστημα:

$$\begin{cases} x+\mathrm{a}y=1 \\ bx-3y=2 \end{cases}$$

(%i12) solve([x+a*y=1,b*x-3*y=2],[x,y]);

(%o12) $[[x=\dfrac{2\mathrm{a}+3}{\mathrm{a}b+3}, y=\dfrac{b-2}{\mathrm{a}b+3}]]$

Θα λύσουμε και ένα μη-γραμμικό σύστημα:

$$\begin{cases} x+yz=2 \\ y-xz=0 \\ x+y=2 \end{cases}$$

(%i13) x+y*z=2y-x*z=0;x+y=2

(%o13) $x+yz=2$

(%o14) $y-xz=0$

(%o15) $x+y=2$

(%i16) solve([%o14,%o15,%o16],[x,y,z]);

(%o16) [[x=1,y=1,z=1],[x=2,y=0,z=0]]

Η εντολή partfrac αναλύει μία ρητή παράσταση σε μερικά κλάσματα (partial fractions). Για να αναλύσουμε την παράσταση: $\mathrm{a}=\dfrac{x}{x^2-5x+6}$

(%i17) a:x/(x^2-5*x+6);

(%o17) $\frac{x}{x^2-5x+6}$

(%i18) partfrac(a,x);

(%o18) $\frac{3}{x-3}-\frac{2}{x-2}$

Με την εντολή ratsimp (rational + simplify) μπορούμε να κάνουμε το αντίστροφο. Ο τελεστής αυτός απλοποιεί μία ρητή παράσταση:

(%i19) ratsimp(%);

(%o19) $\frac{x}{x^2-5x+6}$

Η εντολή funcsolve επιχειρεί να λύσει μια αναδρομική εξίσωση, εφόσον υπάρχει πολυωνυμική ή ρητή λύση. Λύνουμε την $(n+1)f(n)-\frac{n+3}{n+1}f(n+1)=\frac{n-1}{n+2}$ ως εξής:

(%i20) eqn:(n+1)*f(n)-(n+3)*f(n+1)/(n+1)=(n-1)/(n+2);

(%o20) $(n+1)f(n)-\frac{(n+3)\,f(n+1)}{n+1}=\frac{n-1}{n+2}$

(%i21) funcsolve(eqn,f(n));

Dependent equations eliminated: (4 3)

(%o21) $f(n)=\frac{n}{(n+1)(n+2)}$

Η εντολή gcd λειτουργεί και στην περίπτωση πολυωνύμων, δίνοντας τον μκδ δύο πολυωνύμων. Με την εντολή ezgcd, η οποία αναφέρεται σε έναν διαφορετικό αλγόριθμο υπολογισμού, δίνουμε μία n-άδα πολυωνύμων και το πρόγραμμα επιστρέφει μία (n+1)-άδα, όπου το πρώτο στοιχείο είναι ο μέγιστος κοινός διαιρέτης των n πολυωνύμων και τα υπόλοιπα n στοιχεία είναι οι λόγοι των πολυωνύμων προς τον μέγιστο κοινό διαιρέτη. Ένα παράδειγμα:

(%i22) ezgcd(x^2+3*x+2,x+1);

(%o22) [x+1,x+2,1]

Δηλαδή $\mu\kappa\delta\left(x^2+3x+2,x+1\right)=x+1$, $\frac{x^2+3x+2}{x+1}=x+2$, $\frac{x+1}{x+1}=1$

Ιδιαίτερο ενδιαφέρον παρουσιάζει η εντολή eliminate, με την οποία μπορούμε να κάνουμε απαλοιφή μιας ή περισσότερων μεταβλητών από ένα σύστημα εξισώσεων. Ας υποθέσουμε ότι θέλουμε να βρούμε την καμπύλη κατά την οποία τέμνονται οι επιφάνειες:

$$\begin{cases} x^2+y^2+z^2=3 \\ xy+yz+zx=-1 \end{cases}$$

(%i23) a:x^2+y^2+z^2-3;b:x*y+y*z+z*x+1;

(%o23) $z^2+y^2+x^2=3$

(%o24) $yz+xz+xy+1=0$

(%i25) eliminate([a,b],[z]);

(%o25) $[y^4+2xy^3+(3x^2-3)y^2+(2x^3-4x)y+x^4-3x^2+1]$

Όριο, παράγωγος και ολοκλήρωμα

Τα όρια υπολογίζονται με τον τελεστή limit. Το όρισμα περιλαμβάνει τη συνάρτηση, την ανεξάρτητη μεταβλητή και το σημείο στο οποίο τείνει η ανεξάρτητη μεταβλητή. Για τον υπολογισμό του γνωστού $\lim_{x\to 0}\frac{\sin x}{x}$ γράφουμε:

(%i1) f(x):=sin(x)/x;

(%o1) $f(x):=\frac{\sin(x)}{x}$

(%i2) limit(f(x),x,0);

(%o2) 1

Για το όριο της συνάρτησης $y=\frac{x}{|x|}$ όταν $x\to 0^+$, δηλώνουμε πρώτα ότι x>0 με την εντολή assume και γράφουμε:

(%i3) assume(x>0);

(%o3) [x>0]

(%i4) limit(x/abs(x),x,o);

(%o4) 1

Ένας άλλος τρόπος να βρούμε πλευρικά όρια είναι να δηλώσουμε το ‘αριστερά’ και ‘δεξιά’ του αριθμού με τα minus και plus. Πρώτα όμως πρέπει να αναιρέσουμε την εντολή %i3 που δώσαμε προηγουμένως:

(%i5) forget(x>0);

(%o5) [x>0]

(%i6) limit(x/abs(x),x,0,plus);

(%o6) 1

(%i7) limit(x/abs(x),x,0,minus);

(%o7) -1

Για το γνωστό όριο $\lim_{x\to\infty}\left(1+\frac{1}{x}\right)^x=e$:

(%i8) h(x):=(1+1/x)^x;

(%o8) $h(x):=\left(1+\frac{1}{x}\right)^{x}$

(%i9) limit(h(x),x,inf);

(%o9) %e

Η παράγωγος συμβολίζεται με τον τελεστή diff. Το όρισμα περιλαμβάνει την συνάρτηση, την ανεξάρτητη μεταβλητή και την τάξη παραγώγισης, η οποία αν παραληφθεί, τίθεται αυτόματα ίση με 1.

(%i10) f:x^x;

(%o10) x^{x}

(%i11) diff(f,x);

(%o11) $x^{x}(\log(x)+1)$

Αν θέλουμε να εξετάσουμε αν η συνάρτηση $y=e^{-t^2}$ αποτελεί λύση της διαφορικής εξίσωσης y”+2xy’+2y=0 γράφουμε:

(%i12) y:exp(-t^2);

(%o12) $\%e^{-t^2}$

(%i13)diff(y,t,2)+2*t*diff(y,t)+2*y;

(%o13) 0

Ο τελεστής integrate χρησιμοποιείται για τον υπολογισμό ολοκληρωμάτων. Το όρισμα περιλαμβάνει 4 θέσεις, τη συνάρτηση, την ανεξάρτητη μεταβλητή και τα 2 άκρα του διαστήματος στο οποίο γίνεται η ολοκλήρωση. Αν τα 2 τελευταία παραληφθούν, η ολοκλήρωση γίνεται αόριστα. Αν το πρόγραμμα δεν είναι σε θέση να υπολογίσει το αόριστο ολοκλήρωμα, επιστρέφει την πληροφορία εισόδου. Σημειώνεται ότι το πρόγραμμα δεν δίνει τη σταθερά c στο αποτέλεσμα της αόριστης ολοκλήρωσης.

Για το ολοκλήρωμα $\int_{1}^{2}\frac{1}{x}=\log 2$:

(%i14) integrate(1/x,x,1,2);

(%o14) log(2)

Για το ολοκλήρωμα $\int \frac{x}{x^4+4}dx = \frac{1}{4}\tan^{-1}\left(\frac{x^2}{2}\right)+c$:

(%i15) integrate(x/(x^4+4),x);

(%o15) $\frac{\operatorname{atan}\left(\frac{x^2}{2}\right)}{4}$

Το πρόγραμμα υπολογίζει και μη-γνήσια (γενικευμένα) ολοκληρώματα. Για τον υπολογισμό του $\int_0^{+\infty} e^{-x^2}dx$ γράφουμε:

(%i16) integrate(exp(-x^2),x,0,inf);

(%o16) $\frac{\sqrt{\%pi}}{2}$

Συνήθεις διαφορικές εξισώσεις

Η εντολή ode2 λύνει συνήθεις διαφορικές εξισώσεις πρώτης ή δεύτερης τάξης (ordinary differential equations - ODE). Το όρισμα αποτελείται από 3 στοιχεία: την διαφορική εξίσωση, την εξαρτημένη μεταβλητή και την ανεξάρτητη μεταβλητή. Στο αποτέλεσμα, το πρόγραμμα συμβολίζει με %c τη σταθερά (για τάξη-1) και με %k1, %k2 τις σταθερές (για τάξη-2).

Οι μέθοδοι που εφαρμόζει το Maxima είναι: γραμμικές, χωριζόμενων μεταβλητών, ακριβείς, ομογενείς, Bernoulli για τάξη-1 και σταθερών συντελεστών, ακριβείς, γραμμικές ομογενείς που μετασχηματίζονται σε σταθερών συντελεστών, Euler για τάξη-2, καθώς και μερικές ακόμη κατηγορίες.

Με ‘diff(y,x) συμβολίζουμε την παράγωγο του y ως προς x, dy/dx.

Για παράδειγμα οι εξισώσεις $\frac{dx}{dy} - \frac{x}{y} = 0$ και $y'' - 3y' + 2y = 0$:

(%i1) ode2('diff(y,x)-x/y=0,y,x);

(%o1) $\frac{y^2}{2} = \frac{x^2}{2} + \%c$

(%i2) ode2('diff(y,x,2)-3*'diff(y,x)+2*y=0,y,x);

(%o2) $y = \%k1\ \%e^{2x} + \%k2\ \%e^{x}$

Γραμμική άλγεβρα

Ας ορίσουμε ένα μητρώο (πίνακα) στο Maxima, με την εντολή entermatrix, καθορίζοντας την διάστασή του:

(%i1) entermatrix(2,2);

Is the matrix 1. Diagonal 2. Symmetric 3. Antisymmetric 4. General

Answer 1, 2, 3 or 4 :

Δηλώνουμε με τον αντίστοιχο αριθμό αν θα ορίσουμε διαγώνιο, συμμετρικό, αντισυμμετρικό ή τυχαίο μητρώο. Δίνουμε την απάντηση 4 και στη συνέχεια το πρόγραμμα μας ζητάει τα στοιχεία του μητρώου, αρχίζοντας από την πρώτη γραμμή (row) μέχρι να καλύψουμε όλες τις στήλες (column):

Row 1 Column 1: 1;

Row 1 Column 2: -2;

Row 2 Column 1: -3;

Row 2 Column 2: 7;

Matrix entered.

(%o1) $\begin{pmatrix} 1 & -2 \\ -3 & 7 \end{pmatrix}$

Το πρόγραμμα δείχνει το μητρώο που δηλώθηκε. Με τις εντολές transpose, determinant και invert, βρίσκουμε το ανάστροφο, την ορίζουσα και το αντίστροφο. Ας συμβολίσουμε με m το μητρώο:

(%i2) m:%o1;

(Αυτό θα μπορούσε εξ αρχής να δηλωθεί με την εντολή m:entermatrix(2,2);)

(%o2) $\begin{pmatrix} 1 & -2 \\ -3 & 7 \end{pmatrix}$

(%i3) transpose(m);

(%o3) $\begin{pmatrix} 1 & -3 \\ -2 & 7 \end{pmatrix}$

(%i4) determinant(m);

(%o4) 1

(%i5) invert(m);

(%o5) $\begin{pmatrix} 7 & 2 \\ 3 & 1 \end{pmatrix}$

Με την εντολή invert(m),detout; το πρόγραμμα δίνει το αντίστροφο μητρώο, κρατώντας έξω από το μητρώο τον όρο 1/det(m).

Ο πολλαπλασιασμός μητρώων δηλώνεται με την τελεία. Μπορούμε να ορίσουμε 2 ή περισσότερα μητρώα (ακολουθώντας την ανωτέρω διαδικασία) και να εκτελέσουμε πράξεις με αυτά. Σημειώνεται ότι η πρόσθεση μητρώου και αριθμού προσθέτει τον αριθμό σε κάθε στοιχείο του μητρώου.

Με τις εντολές eigenvalues και eigenvectors βρίσκουμε ιδιοτιμές και ιδιοδυανύσματα. Όταν εφαρμοστεί η εντολή eigenvalues, το πρόγραμμα εξάγει 2 n-άδες (n η διάσταση του τετραγωνικού πίνακα). Η πρώτη δηλώνει τις αριθμητικές τιμές των ιδιοτιμών και η δεύτερη την πολλαπλότητα της αντίστοιχης ιδιοτιμής.

(%i6) eigenvalues(m);

(%o6) $\left[\left[4-\sqrt{15},\sqrt{15}+4\right],[1,1]\right]$

Γραφήματα

Το Maxima σχεδιάζει γραφήματα 2 ή 3 διαστάσεων, με τις εντολές plot2d και plot3d. Μέσα στην εντολή περιλαμβάνουμε το τμήμα του x-άξονα που θα σχεδιαστεί.

(%i1) plot2d(x^2-x+3,[x,-10,10]);

και με 3 διαφορετικές συναρτήσεις στο ίδιο γράφημα:

(%i2) plot2d([x^2,x^3,x^4-x+1] ,[x,-10,10]);

Για γράφημα σε 3 διαστάσεις:

(%i3) f(x,y):= sin(x)+cos(y);

(%i4) plot3d(f(x,y),[x,-5,5],[y,-5,5]);