Περιήγηση στο Sage

Δημοσίευση 10.4.rc1

The Sage Development Team

27 Ιουνίου 2024

Περιεχόμενα

Αυτή είναι μία σύντομη περιήγηση στο Sage και στην χρήση του ως αριθμομηχανή.

Η γραμμή εντολών στο Sage εκκινεί με το μήνυμα προτροπής «`sage:`». Για πειραματισμό με τα ακόλουθα παραδείγματα, αρκεί να εισαγάγετε το μέρος μετά το μήνυμα προτροπής.

```
sage: 3 + 5
8
```

Εάν χρησιμοποιείτε το Sage σε σημειωματάριο Jupyter, τότε – παρομοίως – τοποθετείστε τα πάντα έπειτα του μηνύματος προτροπής εντός ενός κελιού εισαγωγής, και πατήστε `Shift-Enter` για να λάβετε την αντίστοιχη έξοδο.

Το σύμβολο εκθέτη σημαίνει «ύψωση σε δύναμη».

```
sage: 57.1^100
4.60904368661396e175
```

Υπολογίζουμε τον αντίστροφο ενός 2×2 πίνακα στο Sage.

```
sage: matrix([[1, 2], [3, 4]])^(-1)
[  -2    1]
[ 3/2 -1/2]
```

Εδώ υπολογίζουμε το ολοκλήρωμα μίας απλής συνάρτησης.

```
sage: x = var('x')   # δημιουργίας συμβολικής μεταβλητής
sage: integrate(sqrt(x) * sqrt(1 + x), x)
1/4*((x + 1)^(3/2)/x^(3/2) + sqrt(x + 1)/sqrt(x))/((x + 1)^2/x^2 - 2*(x + 1)/x + 1)
- 1/8*log(sqrt(x + 1)/sqrt(x) + 1) + 1/8*log(sqrt(x + 1)/sqrt(x) - 1)
```

Εδώ το Sage καλείται να λύσει μία δευτεροβάθμια εξίσωση. Το σύμβολο == αντιπροσωπεύει την ισότητα στο Sage.

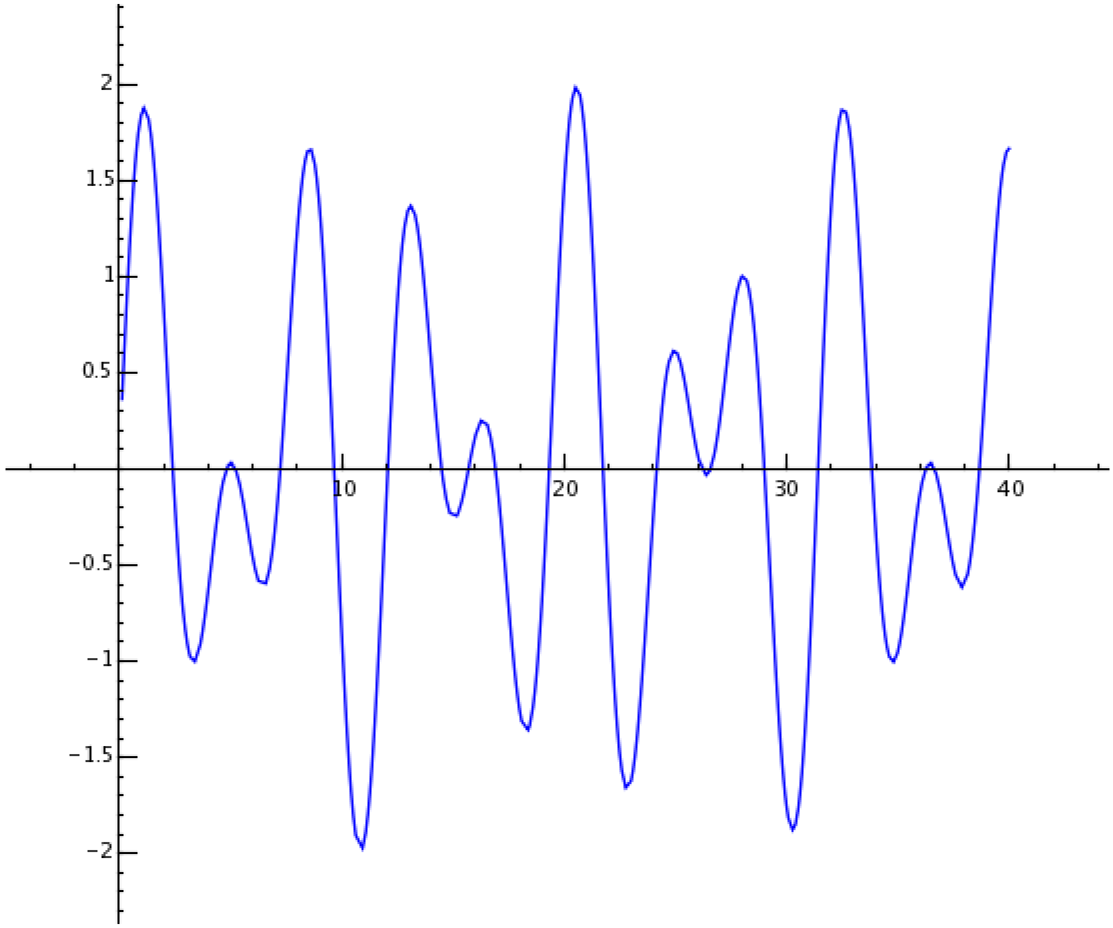
```
sage: a = var('a')
sage: S = solve(x^2 + x == a, x); S
[x == -1/2*sqrt(4*a + 1) - 1/2, x == 1/2*sqrt(4*a + 1) - 1/2]
```

Το αποτέλεσμα είναι μία λίστα από ισότητες.

```
sage: S[0].rhs()  # δεξί μέρος της εξίσωσης (rhs = right hand side)
-1/2*sqrt(4*a + 1) - 1/2
```

Το Sage μπορεί να παραγάγει γραφήματα για διάφορες συναρτήσεις.

```
sage: show(plot(sin(x) + sin(1.6*x), 0, 40))
```

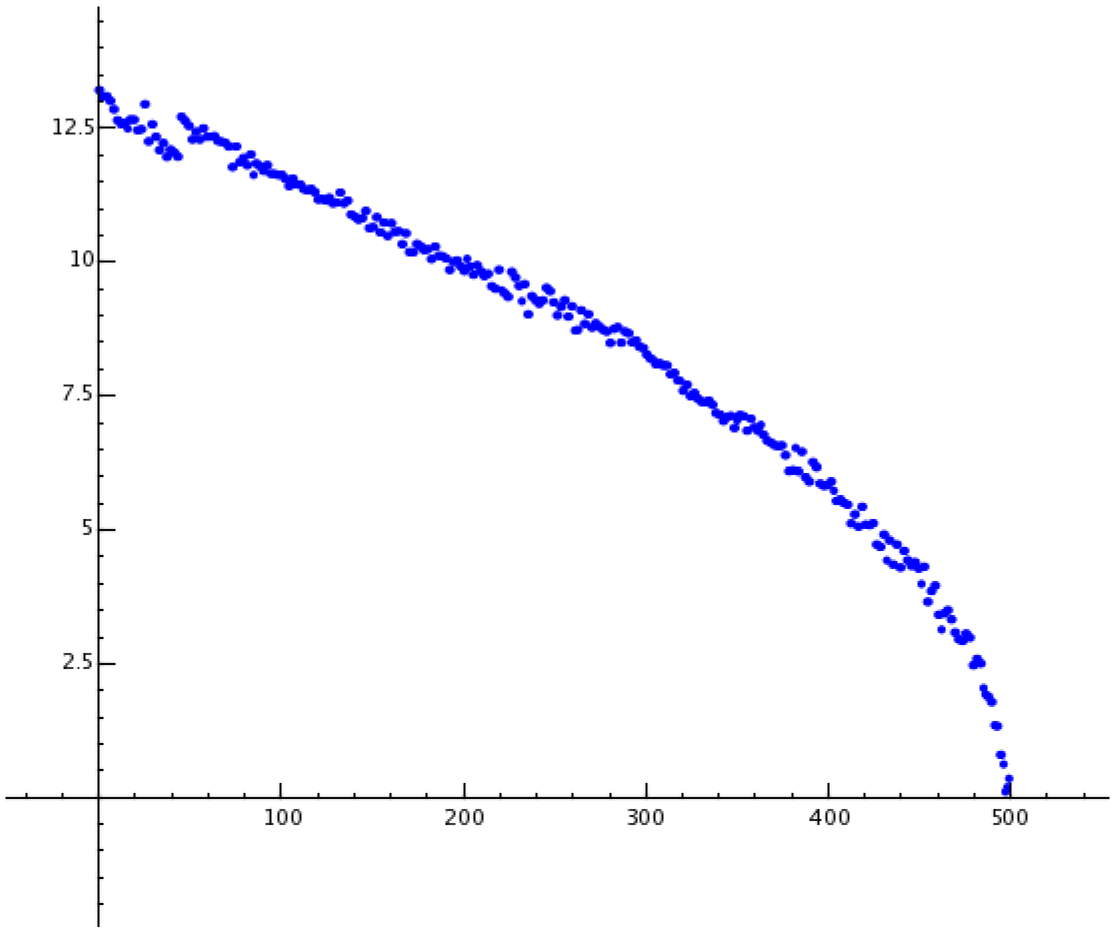


Το Sage είναι μία πολύ ισχυρή αριθμομηχανή. Για να το δείτε αυτό, δημιουργείστε έναν 500×500 πίνακα με τυχαίους αριθμούς.

```
sage: m = random_matrix(RDF, 500)
```

Το Sage χρειάζεται ένα δευτερόλεπτο για τον υπολογισμό και την γραφική παρουσίαση των ιδιοτιμών του πίνακα.

```
sage: e = m.eigenvalues()  # περίπου 1 δευτερόλεπτο
sage: w = [(i, abs(e[i])) for i in range(len(e))]
sage: show(points(w))
```



Το Sage μπορεί να διαχειριστεί τεράστιους αριθμούς, ακόμα και με εκατομμύρια ή δισεκατομμύρια ψηφία.

```
sage: factorial(100)
9332621544394415268169923885626670049071596826438162146859296389521759999322991560894146397615651828
```

```
sage: n = factorial(1000000)   # περίπου 1 δευτερόλεπτο
sage: len(n.digits())
5565709
```

Εδώ υπολογίζουμε τουλάχιστον 100 ψηφία του αριθμού π.

```
sage: N(pi, digits=100)
3.
↪14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706
```

Εδώ το Sage παραγοντοποιεί ένα πολυώνυμο δύο μεταβλητών.

```
sage: R.<x,y> = QQ[]
sage: F = factor(x^99 + y^99)
sage: F
(x + y) * (x^2 - x*y + y^2) * (x^6 - x^3*y^3 + y^6) *
(x^10 - x^9*y + x^8*y^2 - x^7*y^3 + x^6*y^4 - x^5*y^5 +
 x^4*y^6 - x^3*y^7 + x^2*y^8 - x*y^9 + y^10) *
```

(συνέχεια στην επόμενη σελίδα)

(συνεχίζεται από την προηγούμενη σελίδα)

```
(x^20 + x^19*y - x^17*y^3 - x^16*y^4 + x^14*y^6 + x^13*y^7 -
 x^11*y^9 - x^10*y^10 - x^9*y^11 + x^7*y^13 + x^6*y^14 -
 x^4*y^16 - x^3*y^17 + x*y^19 + y^20) * (x^60 + x^57*y^3 -
 x^51*y^9 - x^48*y^12 + x^42*y^18 + x^39*y^21 - x^33*y^27 -
 x^30*y^30 - x^27*y^33 + x^21*y^39 + x^18*y^42 - x^12*y^48 -
 x^9*y^51 + x^3*y^57 + y^60)
sage: F.expand()
x^99 + y^99
```

Το Sage χρειάζεται λιγότερο από 1 δευτερόλεπτο για να υπολογίσει τους τρόπους με τους οποίους ο αριθμός 100 εκατομμύρια μπορεί να γραφεί ως άθροισμα θετικών ακεραίων.

```
sage: z = Partitions(10^8).cardinality()  # περίπου .1 δευτερόλεπτα
sage: z
1760517045946249141360373894679135204009...
```

Το Sage είναι το πιο προηγμένο λογισμικό ανοιχτού κώδικα για μαθηματικά στον κόσμο.