SageMath

SageMath, uvod 1: šta je to CAS?

- treba predavati u **prvoj** godini studija (još jedna jeretička misao)
- symbolic computation, algebraic computation, computer algebra . . .
- nekad smatrano za umnu delatnost . . .
- u osnovi manipulacija stringova po utvrđenim pravilima . . .
- sećate se zbirke Demidovič?
- sve ovo može da se automatizuje i automatizuje se odavno, Macsyma početa 1968, . . .
- computer algebra systems (CAS)? `http://en.wikipedia.org/wiki/Comparison_of_computer_algebra_systems`

SageMath, uvod 2: CAS proprietary alternative

- **Mathematica**, $2,495 (Professional), $1095 (Education), $140 (Student), $69.95 (Student annual license), $295 (Personal), **free on Raspberry Pi hardware**
- **Maple**, $2,390 (Commercial), $2,265 (Government), $995 (Academic), $239 (Personal Edition), $99 (Student), $79 (Student, 12-Month term)
- navijačke strasti, Maple vs. Mathematica
- Symbolic Math Toolbox (MATLAB), $3,150 (Commercial), $99 (Student Suite), $700 (Academic), $194 (Home) including required Matlab
- bio popularan **Derive**, `http://en.wikipedia.org/wiki/Derive_(computer_algebra_system)`, discontinued 2007

SageMath, uvod 3: CAS slobodne alternative

- Maxima, a computer algebra system, GPL
- `http://en.wikipedia.org/wiki/Maxima_(software)`
- `http://maxima.sourceforge.net/`
- zasnovana na MIT Macsyma, `http://en.wikipedia.org/wiki/Macsyma`
- William Frederick Schelter, GPL, DOE Macsyma 1982 `http://en.wikipedia.org/wiki/Bill_Schelter`
- star program, ali aktivno se razvija
- uključen u **Scilab** i **Euler Math Toolbox**
- uključen u **SageMath**, `http://en.wikipedia.org/wiki/Sage_(mathematics_software)`
- **napredno: SageMath**, `http://www.sagemath.org/`

SageMath, uvod 4: CAS slobodne alternative

- SymPy paket za simboličko računanje koristeći Python
- cilj: isto što i wxMaxima
- ideja: Python sintaksa, poznata
- moduli, funkcije za simboličko računanje
- `http://sympy.org/en/index.html`
- `https://github.com/sympy/sympy/releases`
- `http://live.sympy.org/`
- uputstvo 1.7.1, 2592 strane, 11.12.2020; čitate stranu po stranu?
- usput, uputstvo 1.3, 2044 strane, 14.09.2018; napreduje?
- Ubuntu: Software Center ili Synaptic
- volite Mathematica sintaksu?
- **odličan pregled i tutorial**
- **za one koji vole matematiku:** `http://www.sagemath.org/`

SageMath, uvod 5: SageMath, konačno

- `http://www.sagemath.org/`
- od 24.02.2005, GPL
- Python sintaksa
- koristi Maxima i SymPy i mnoge druge free programe, kombinuje ih
- zato je napuštena politika dva alata za jedan posao
- **Literatura:**
  1. Mathematical Computation with Sage
  2. Sage for Undergraduates
- za Ubuntu 20.04 uključen u repository
- koristi pun kapacitet free software ideologije, kombinuje puno različitih alata u novu funkcionalnost, izbegava "reinventing the wheel"
- autor (pokretač) programa: William A. Stein

SageMath: početak sa komandne linije

- za moj uzrast i vid na manje od 1 m bilo je zgodno prometiti color scheme terminala na light
- `sage`
- pogledajte ime terminala, IPython
- `whos`
- `3 + 2`
- `factorial(100)`
- `x = factorial(100)`
- `type(x)`
- `whos`
- `del x`
- `whos`

SageMath: faktorizacija celih brojeva, ^ i **

- `factor(123456789)`
- `_`
- `type(_)`
- `_`
- `factor(123456789)`
- `expand(_)`
- `type(_)`
- zaista dynamic typing!
- `3^2`
- `3**2`

SageMath: polinomi

- `type(x)`
- `x = var(’x’)`
- `type(x)`
- `p = x^2 + 2 * x + 1`
- `p`
- `show(p)`
- `print(p)`
- `pretty_print(p)`
- `latex(p)`
- `factor(p)`
- `p.factor()`
- `p.roots()`

SageMath: jupyter

- izađete iz sage, ^d (ctrl/d)
- na komandnoj liniji:
- `sage -n jupyter`
- sačekate da se otvori browser, New, SageMath 9.0
- alternativa, na komandnoj liniji
- `jupyter-notebook`
- dalje isto . . .
- modernija alternativa, na komandnoj liniji
- `jupyter lab`
- dalje slično, drugačiji interfejs
- server se gasi sa ctrl/c (^c) na komandnoj liniji ili sa Quit u osnovnom jupyter prozoru
- shortcut keys, da se pojave h
- ponavljamo dosadašnje . . .

SageMath: razlomci, konstante i numerika 1

```
1/2

type(_)

10/26

z = 1/2 - 1/3; z

zz = 1./2 - 1/3; print(zz); print(type(zz))

show(z)

type(z)

(z).n(digits = 10)

(z).n(digits = 100)
```

SageMath: razlomci, konstante i numerika 2

```
i^2

I^2

pi

show(_)

pi.n()

e^(i * pi) + 1

e.n(digits = 100)

pi.n(digits = 500)
```

SageMath: razlomci, konstante i numerika 3

```
r = e^(pi * sqrt(163))

r.n(digits = 27)

r.n(digits = 100)

n(sqrt(2))

n(sqrt(2), digits = 50)

N(sqrt(2))

(sqrt(2)).numerical_approx()

numerical_approx(sqrt(2))
```

SageMath: razlomci, konstante i numerika 4

```
infinity

Infinity

oo

1 / oo

2 + oo
```

SageMath: algebra 1

```
show(((x + 1)^2).expand())

show(((x + 1) * (x - 1)).expand())

y = ((x-3)^7).expand(); show(y)

show(factor(y))

eq = ((x - 4) * (x - 5) * (x - 6) * 7).expand()
show(eq)

seq = solve(eq, x); show(seq)

seq[0].rhs(), seq[1].rhs(), seq[2].rhs()

show(factor(eq))
```

SageMath: algebra 2

```
show(factor(4 * x^5 - 4 * x^4  -
13 * x^3 + x^2 - 17 * x + 5))

s = var(’s’)
ex = (s^3 + 4 * s^2 + 6 * s + 4) / \
(s^3 + 3 * s^2 + 3 * s + 1)
show(ex)

show(factor(ex))

pfex = ex.partial_fraction(); show(pfex)

show(pfex.expand())

show(pfex.simplify_rational())
```

SageMath: algebra 3

```
expression = sin(x / (x^2 + x)) == \
exp((log(x) + 1)^2 - log(x)^2); show(expression)

show(expression.simplify())

show(expression.expand())

show(expression.simplify_rational())
```

SageMath: algebra 4

```
a = var(’a’)
expr = (x^(a / 2) + 1)^2 * (x^(a / 2) - 1)^2 \
/ (x^a - 1)
show(expr)

show(expr.expand())

show(expr.simplify())

show(expr.rational_simplify())

show(expr.simplify_full())
```

SageMath: trigonometrija 1

```
cos(pi / 3)

show(sin(pi / 3))

sin(pi / 3).n()

sin(pi / 3.)

sin(1)

sin(1.)
```

SageMath: trigonometrija 2

```
show(csc(45 * pi / 180))

tan(pi / 4)

show(tan(pi / 8))

show(tan(pi / 16))

show(tan(pi / 32))

show(acos(1/2))

180 / pi * asin(sqrt(3) / 2)

show(acsc(1))
```

SageMath: trigonometrija 3

```
eq = sin(x)^2 + cos(x)^2; print(eq)
show(eq)
latex(eq)

eq.simplify()

eq.trig_simplify()

a, b = var(’a b’)
e1 = sin(a + b)
show(e1)
```

SageMath: trigonometrija 4

```
e2 = e1.trig_simplify(); show(e2)

e2.trig_reduce()

e3 = sin(x)^2; show(e3)

e4 = e3.trig_reduce(); show(e4)

show(e4.simplify_full())
```

SageMath: jednačine 1

```
a, b, c = var(’a b c’)
eq = a * x^2 + b * x + c
s = solve(eq, x)
show(s)

x1 = s[0].rhs(); show(x1)

x2 = s[1].rhs(); show(x2)

eq.subs(x = x1)

_.expand()

eq.subs(x = x2).expand()
```

SageMath: jednačine 2

```
s2 = solve(x^3 + 4 * x^2 - 3 * x + 1, x); show(s2)

latex(s2)

n(s2[0].rhs()), n(s2[1].rhs()), n(s2[2].rhs())

show(n(s2[1].rhs()))

y = var(’y’)
solve([x + y == 3, x - y == 1], [x, y])

solve([x + y == 2, 2 * x + 2 * y == 4], [x, y])

solve([x + y == 2, 2 * x + 2 * y == 5], [x, y])
```

SageMath: algebarske jednačine, sistemi

```
x, y = var(’x y’)
e1 = x^2 + y^2 == 41
e2 = y == x + 1
show(e1)
show(e2)

s = solve([e1, e2], [x, y])
show(s)

s1 = solve(e1, y); show(s1)

y1 = s1[0].rhs(); show(y1)

y2 = s1[1].rhs(); show(y2)

set_verbose(-1)
plot([y1, y2, e2.rhs()], (x, -10, 10), aspect_ratio = 1)
```

SageMath: limesi 1

```
limit((1 + 1 / x)^(3 * x), x = infinity)

_.n()

show(__)

f = (x - 2) / (x^2 - 4)
#f(x = 2)
limit(f, x = 2)

show(limit(x^3, x = oo))

show(limit(x^3, x = - oo))
```

SageMath: limesi 2

```
show(limit(atan(x), x = oo))

show(limit(atan(x), x = - oo))

show(limit(sin(17 * x) / x, x = 0))

show(limit(1 / x, x = 0))

show(limit(1 / x, x = 0, dir = ’+’))

show(limit(1 / x, x = 0, dir = ’-’))

f(x) = atan(x)
h = var(’h’)
show(limit((f(x + h) - f(x)) / h, h = 0))
```

SageMath: izvodi 1

```
diff(x^2, x)

(x^2).diff()

diff(x^2, x, 2)

(x^2).diff(x, 2)

(x^2).diff(2)

(x^2).diff(x)

diff(sin(x))

diff(sin(x), 2)
```

SageMath: izvodi 2

```
diff(sin(x), x, 3)

(sin(x)).diff(4)

show((atan(x)).diff(x))

diff((x^2 + 2 * x + 1) * e^(3 * x), x)

_.factor()

show(_)

y = var(’y’)

diff(sin(x * y), x)

diff(sin(x * y), y)
```

SageMath: integrali 1

```
integrate(sin(x), x)

integrate(sin(x), (x, 0, pi))

integrate(1/(1 + x^2), x)

integrate(1/(1 + x^2), (x, 0, 1))

show(_)
```

SageMath: integrali 2

```
show(integrate((3 * x + 5) / (x^2 + x + 1), x))

integrate(x^-(1/5), (x, 4, 5))

show(_)

(__).n()

integrate((log(x) / x)^2, (x, 1, oo))
```

SageMath: kompleksni brojevi 1

```
z = 3 + 4j; show(z); show(type(z))

Z = 3 + 4 * i; show(Z); show(type(Z))

abs(z)

show(abs(Z)); show(arg(Z))

x, y = var(’x, y’)

z = x + i * y; show(z)

print(z.real()); show(z.real());
print(z.imag()); show(z.imag())
```

SageMath: kompleksni brojevi 2

```
show(z.conjugate())

w = 1 / z; show(w)

show(w.rectform())

show(abs(z)); show(abs(w))

show(z.norm()); show(w.norm());
show((w.norm()).simplify_full())
```

SageMath: Laplace 1

```
t = var(’t’)
s = var(’s’)
a = var(’a’)
w = var(’w’)

show(laplace(exp(- a * t), t, s))

show(laplace(sin(w * t), t, s))

show(laplace(cos(w * t), t, s))
```

SageMath: Laplace 2

```
inverse_laplace(s / (s^2 + 1), s, t)

inverse_laplace(1 / (s^2 + 1), s, t)

inverse_laplace(1 / s, s, t)

inverse_laplace(1 / s^2, s, t)

inverse_laplace(1 / (s + 1), s, t)
```

SageMath: Laplace 3

```
inverse_laplace(1, s, t)

ex = (s^3 + 4 * s^2 + 6 * s + 4) / \
(s^3 + 3 * s^2 + 3 * s + 1)
show(ex)

show(inverse_laplace(ex * 1 / s, s, t))
```

SageMath: diferencijalne jednačine 1

```
t = var(’t’)
y = function(’y’)(t)

deq1 = diff(y, t) + y == 0
show(deq1)
desolve(deq1, [y, t])
print(desolve(deq1, [y, t])
show(desolve(deq1, [y, t])

deq2 = diff(y, t) - y == 0
show(deq2)
show(desolve(deq2, [y, t]))

deq3 = diff(y, t) + y == 1
show(deq3)
s3 = desolve(deq3, [y, t])
print(s3); show(s3)
```

SageMath: diferencijalne jednačine 2

```
print(expand(s3))
show(expand(s3))

deq4 = diff(y, t) + y == cos(2 * t)
show(deq4)
desolve(deq4, [y, t])

expand(_)

sol = _

show(sol)

deq5 = diff(y, t, 2) + 4 * y == 0
show(deq5)
show(desolve(deq5, [y, t]))
```

SageMath: diferencijalne jednačine 3

```
deq6 = diff(y, t, 2) - 4 * y == 0
show(deq6)
show(desolve(deq6, [y, t]))

deq7 = diff(y, t, 2) + 4 * y == cos(2 * t)
show(deq7)
show(desolve(deq7, [y, t]))

deq8 = diff(y, t, 2) + 4 * y == cos(t)
show(deq8)
show(desolve(deq8, [y, t]))

deq9 = diff(y, t, 2) + 2 * diff(y, t) + 4 * y == cos(t)
show(deq9)
show(desolve(deq9, [y, t]))
```

SageMath: diferencijalne jednačine 3

```
deq10 = diff(y, t, 2) + 4 * diff(y, t) + 4 * y == cos(t)
show(deq10)
show(desolve(deq10, [y, t]))

deq11 = diff(y, t, 2) + 4 * diff(y, t) + 4 * y == cos(t)
show(deq11)
show(desolve(deq11, [y, t]))
```

so, we're done . . .

šta nismo stigli, a bilo je u planu punom optimizma?

- SageMath, plotovanje
- SageMath, linearna algebra (za ovo mi je baš žao)
- **Julia** (za ovo mi je najviše žao)
- samo ovo nije kraj, idite dalje, samo sada idete sami . . .
- nadam se da ste kroz PSAE prošli početne korake i postali slobodni i samostalni
- happy hacking!

šta nismo ni planirali, a potrebno je, $t \ll$ (potrebnog)

- **komandna linija**
- **regular expressions**, Charles Severance, video, 35', 23''
- detaljnije Code::Blocks ili neki drugi IDE
- **Eclipse** `http://www.eclipse.org/`
- ukratko LibreOffice, više pravila pisanja
- makar malo vremena za **SciLab**
- malo vremena za **GIMP**, mada nije problem
- malo vremena za **Inkscape**, ovo je veći problem
- **Ngspice** `http://ngspice.sourceforge.net/`
- **moj izbor:** julia
- **ovo je samo početak, mada je za sada . . .**

— K R A J —