Linear Algebra Step by Step in SymPy

Dario Beraldi

March 14, 2015

Contents

Abstract

Excerpts from *Linear Algebra Step by Step* by K. Singh with SymPy implementation.

Linear Equations and Matrices

Solving linear systems

Exercises 1.1 2f

$$\begin{aligned} ex - ey &= 2 \\ ex + ey &= 0 \end{aligned} \tag{1}$$

```
x, y= symbols('x y', real= true)
eq1= Eq(E*x - E*y, 2)
eq2= Eq(E*x + E*y, 0)
sols= solve([eq1, eq2])
```

Solved for:

$$\left\{x : e^{-1}, \quad y : -\frac{1}{e}\right\} \tag{2}$$

Check solutions by substituting them in the original equations 1

```
eq1.subs(sols)
True
eq2.subs(sols)
True
```

Solve the system 1 using matrix notation

```
coeffs= Matrix([[E, -E], [E, E]])
const= Matrix([2, 0])
[coeffs, const]
```

$$\left[\begin{bmatrix} e & -e \\ e & e \end{bmatrix}, \quad \begin{bmatrix} 2 \\ 0 \end{bmatrix}\right] \tag{3}$$

If everything is correct, solutions are consistent with 2:

```
coeffs.solve(const)
```

$$\begin{bmatrix} e^{-1} \\ -\frac{1}{e} \end{bmatrix} \tag{4}$$

Exercises 1.1 4a

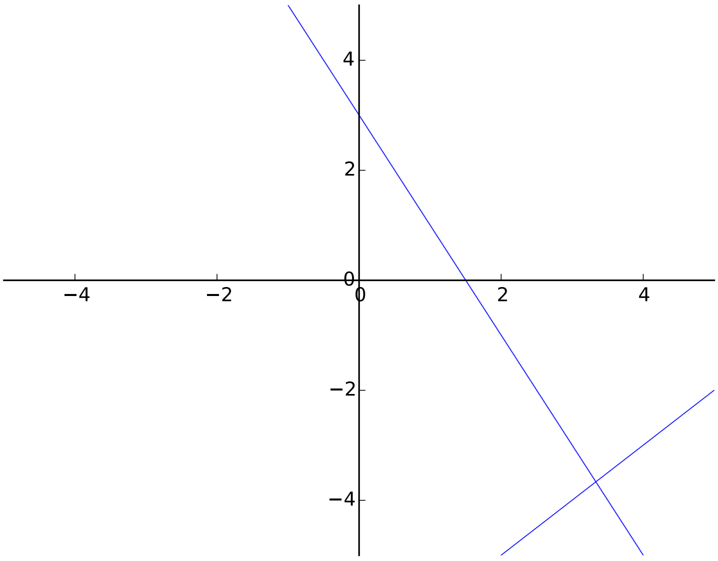
Plot the graphs of these linear equations

$$\begin{aligned} 2x + y = 3 \\ x - y = 7 \end{aligned} \tag{5}$$

```
eq1= Eq(2*x + y, 3)
eq2= Eq(x - y, 7)
```

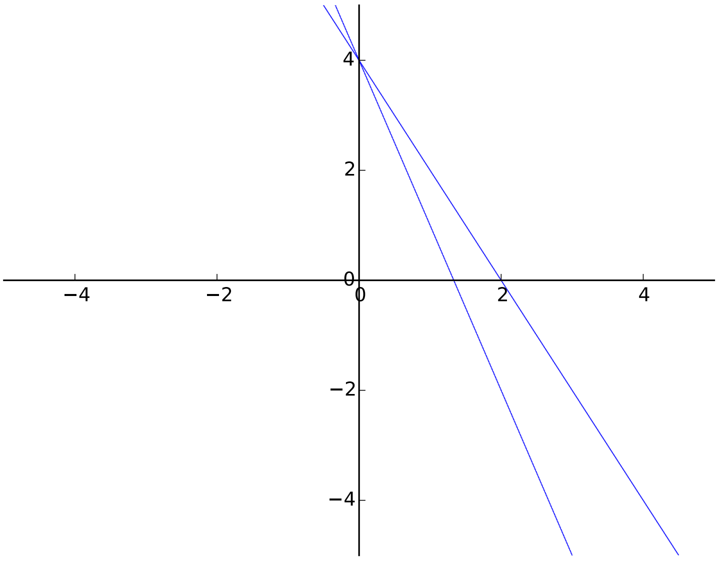
Plot

```
pp1= plot_implicit(eq1)
pp2= plot_implicit(eq2)
pp1.extend(pp2)
pp1.save('figs/ex4a.pdf')
```



Exercises 1.1 5b

```
eq1= Eq(12*x + 4*y, 16)
eq2= Eq(8*x + 4*y, 16)

pp1= plot_implicit(eq1)
pp2= plot_implicit(eq2)
pp1.extend(pp2)
pp1.save('figs/ex5b.pdf')
```



Solve by row echelon form

Given a linear system of equations, the solutions can be found by Gaussian elimination:

- Extract the coefficients and constants from the equations and put them in an augmented matrix.
- Transform the augmented matrix in reduced row echelon form (rref). This is the result of the Gaussian elimination process.
- The last columns of the rref matrix lists the solutions of the linear system.

Reminder: rref means that every leading coefficient is 1 and is the only nonzero entry in its column [1].

Exercises 1.2 2a

$$\begin{aligned} x + 2y + 3z &= 12 \\ 2x - y + 5z &= 3 \\ 3x + 3y + 6z &= 21 \end{aligned} \tag{6}$$

```
eq1= Eq(x + 2*y + 3*z, 12)
eq2= Eq(2*x - y + 5*z, 3)
eq3= Eq(3*x + 3*y + 6*z, 21)
```

[1] http://en.wikipedia.org/wiki/Row_echelon_form

Represent the linear systen as *augmented* matrix, where the last column holds the constants:

$$M = \begin{bmatrix} 1 & 2 & 3 & \mathit{12} \\ 2 & -1 & 5 & \mathit{3} \\ 3 & 3 & 6 & \mathit{21} \end{bmatrix} \tag{7}$$

In SymPy use the `Poly` class to conveniently extract the coefficients at the left hand side of the equations:

```
M= Matrix([
    Poly(eq1.lhs).coeffs(),
    Poly(eq2.lhs).coeffs(),
    Poly(eq3.lhs).coeffs()
])
const= Matrix([eq1.rhs, eq2.rhs, eq3.rhs])
M= const.col_insert(0, M)
```

In reduced row echelon form, with indexes of the pivot variables on the right:

$$RREF = \left(\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0, & 1, & 2 \end{bmatrix} \right) \tag{8}$$

Solutions can be read from last column of the RREF $\begin{bmatrix} x:1 \\ y:4 \\ z:1 \end{bmatrix}$. Verify the solutions solve the initial system of equations:

```
rref= M.rref()
sols= rref[0].col(-1)

eq1.subs({x:sols[0], y:sols[1], z:sols[2]}) # True
eq2.subs({x:sols[0], y:sols[1], z:sols[2]}) # True
eq3.subs({x:sols[0], y:sols[1], z:sols[2]}) # True

# Or the same returning dict {x: 1, y: 4, z: 1}
solve([eq1, eq2, eq3])
```

Exercises 1.2 3d

Linear system:

$$\begin{aligned} -2x + 3y - 2z &= 8 \\ -x + 2y - 10z &= 0 \\ 5x - 7y + 4z &= -20 \end{aligned} \tag{9}$$

Augmented matrix:

$$\begin{bmatrix} -2 & 3 & -2 & 8 \\ -1 & 2 & -10 & 0 \\ 5 & -7 & 4 & -20 \end{bmatrix} \tag{10}$$

Reduced row echelon form with solutions in the last column:

$$\left(\begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & \frac{1}{2} \end{bmatrix}, \quad \begin{bmatrix} 0, & 1, & 2 \end{bmatrix} \right) \tag{11}$$

In SymPy :

```
eq1= Eq(-2*x + 3*y -2*z, 8)
eq2= Eq(-x + 2*y - 10*z, 0)
eq3= Eq(5*x - 7*y + 4*z, -20)

M= Matrix([
    Poly(eq1.lhs).coeffs(),
    Poly(eq2.lhs).coeffs(),
    Poly(eq3.lhs).coeffs()
])
const= Matrix([eq1.rhs, eq2.rhs, eq3.rhs])
M= const.col_insert(0, M)
rref= M.rref()
sols= rref[0].col(-1)

# Checked: {x: -3, y: 1, z: 1/2}
solve([eq1, eq2, eq3])
```

Vector arithmetic

Exercise 1.3.1

Given two vectors:

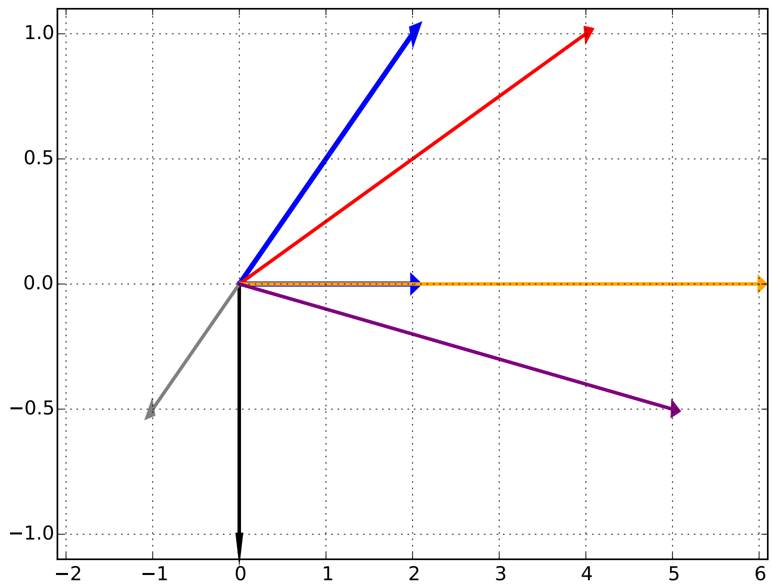
```
va= Matrix([2, 0])
vb= Matrix([2, 1])

fig = plt.figure()
fig.add_subplot(111)
```

plot the results of the operations

(a-e)

```
vA= va + vb          # [4 1]
vB= va - vb          # [0 -1]
vC= 3 * va           # [6 0]
vD= -1/2 * vb        # [-1 -0.5]
vE= 3*va - 1/2 * vb  # [5 -0.5]
```

And plot vectors




```
plt.arrow(0, 0, float(va[0]), float(va[1]), lw= 3, color= 'b', head_width=0.05)
plt.arrow(0, 0, float(vb[0]), float(vb[1]), lw= 3, color= 'b', head_width=0.05)
plt.arrow(0, 0, float(vA[0]), float(vA[1]), lw= 2, color= 'r', head_width=0.05)
plt.arrow(0, 0, float(vB[0]), float(vB[1]), lw= 2, color= 'black', head_width=0.05)
plt.arrow(0, 0, float(vC[0]), float(vC[1]), lw= 2, color= 'orange', head_width=0.05)
plt.arrow(0, 0, float(vD[0]), float(vD[1]), lw= 2, color= 'grey', head_width=0.05)
plt.arrow(0, 0, float(vE[0]), float(vE[1]), lw= 2, color= 'purple', head_width=0.05)
plt.xlim(-2.1, 6.1)
plt.ylim(-1.1, 1.1)
plt.grid()
plt.savefig('figs/ex1_3.pdf')
plt.close()
```

Exercise 1.3.8

Show that $x\mathbf{u} + y\mathbf{v} = \mathbf{w}$ where $\mathbf{u} = (1\ 0)^T$, $\mathbf{v} = (0\ 1)^T$, and $\mathbf{w} = (x\ y)^T$.
This is a consequence of $x(1\ 0) = (x\ 0)$ and $y(0\ 1) = (y\ 0)$. So that $(x\ 0)+(0\ y) = (x\ y)$

```
x, y= symbols('x y')
u= Matrix([1, 0])
v= Matrix([0, 1])
w= Matrix([x, y])
Eq(x*u + y*v, w) # True
```

Exercise 1.3.12

Find the real numbers x, y and z, if

$$\begin{bmatrix} x - 2z \\ 2x + y \\ -y + 6z \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 17 \end{bmatrix} \tag{12}$$

Which is solved for:

$$sols = \left[\left\{x : 7, \quad y : -11, \quad z : 1\right\}\right] \tag{13}$$

```
eq1= Eq(x * Matrix([1, 2, 0]) + y * Matrix([0, 1, -1]) + z * Matrix([-2, 0, 6]),
    Matrix([5, 3, 17]))
sols= solve(eq1)
```

Exercise 1.3.12

Show that vectors $\mathbf{u}$ and $\mathbf{v}$ in space $\mathbb{R}^n$ can be $\mathbf{u} \cdot \mathbf{v} = \mathbf{0}$ even if neither $\mathbf{u}$ or $\mathbf{v}$ are 0 vectors.

Set:

```
u= Matrix([-1, 1])
v= Matrix([1, 1])
u.dot(v) == 0 # True
```