I. Configure Environment

I-a. Install Ipython

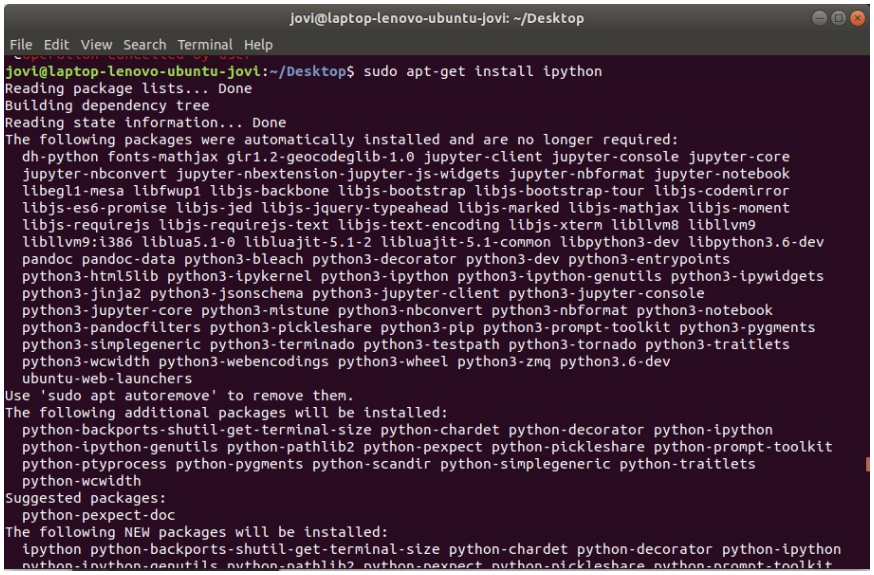

Figure 1: Download from apt source

By the way, I have already installed python 3.6 before.

I-b. Install Jupyter

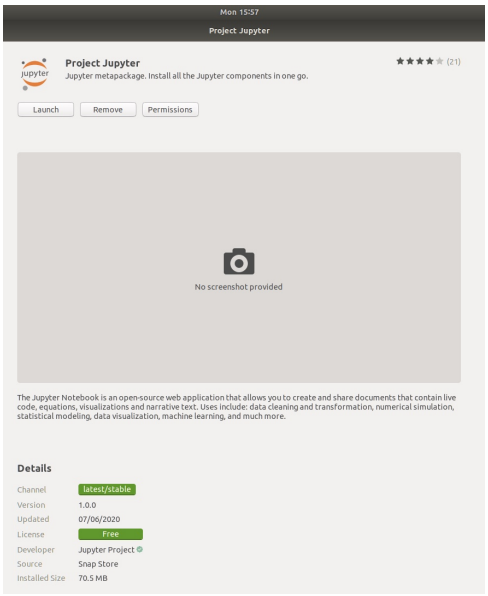

Figure 2: Download from ubuntu software center

But I have some troubles on adding the kernel of octave into this editor, so I determine not to use it.

I-c. Install Octave

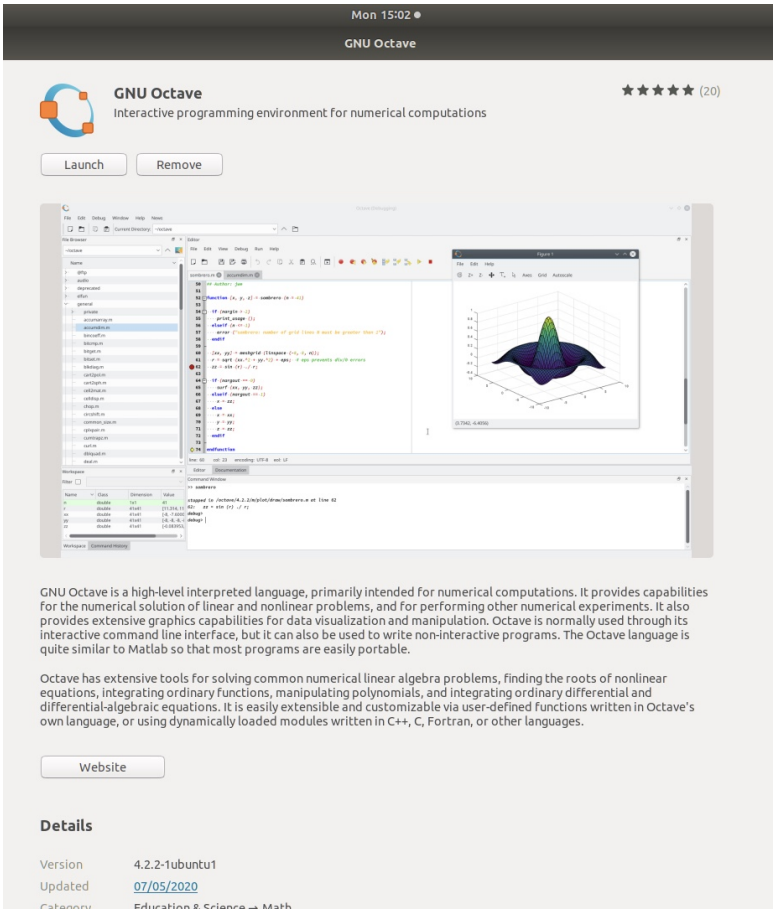

Figure 3: Download from ubuntu software center

Then I install packages such as symbolic, image, optim and statistics via command line in Octave.

II. Problems and Examples

Test Package Symbolic

INPUT: x+3=3x
CODE: root = solve(x+3 == 3*x)
OUTPUT: root = $\frac{3}{2}$

Matrix Multiply

INPUT:
a = [1 2 3]
b = [3 2 1]

CODE: vm = a'*b

OUTPUT:
vm =

3	2	1
6	4	2
9	6	3

Vector Product

INPUT:
a = [1 2 3]
b = [3 2 1]

CODE
ip = dot(a,b)
op = cross(a,b)

OUTPUT
ip = 10
op = -4 8 -4

Initialize Matrix

INPUT
A = rand(3,3)
B = A(2:3, 1:2)
I = ones(4,4)
O = zeros(2,2)

OUTPUT
A =
0.868694705363510 0.431413827463545 0.136068558708664
0.084435845510910 0.910647594429523 0.869292207640089
0.399782649098896 0.181847028302852 0.579704587365570
B =
0.084435845510910 0.910647594429523
0.399782649098896 0.181847028302852
I =
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
O =
0 0
0 0

Newton Iteration

INPUT: f, df, err=1, $x_0 = 2$

CODE
iter = 0;
err = 1;
x0 = 2.0;
x = x0;
format long;
while(err > 1e-8 and iter ¡ 20)

```
x0 = x;
x = x0 - df(x0)/f(x0);
err = norm(x - x0);
iter = iter +1;
fprintf('iter %d: x = %18.15f, f(x) = %28.15f', iter, x, f(x))
end
```

OUTPUT

```
iter 1: x = 1.000000000000000, f(x) = 0.281718171540954
iter 2: x = 0.914155281832543, f(x) = 0.012372566882760
iter 3: x = 0.910017665783406, f(x) = 0.000030034837378
iter 4: x = 0.910007572548888, f(x) = 0.000000000179075
iter 5: x = 0.910007572488709, f(x) = 0.000000000000000
```

Define Function and Plot

INPUT: t = linspace(0,20,40);

CODE

```
plot(t, besselj(0.5, t), 'r*-')
hold on
plot(t,besselj(1.5, t), 'b*–')
plot(t,besselj(5.5,t), 'cs-')
plot(t, besselj(10.5,t), 'mo–')
hold off
function y = myfunc(x)
y = 3 * x.^2 + 2 * x + 18;
end

    function y = df(x)
y = 6 * x - exp(x);
end

    function y = f(x)
y = 3 * x.^2 - exp(x);
end
```

OUTPUT

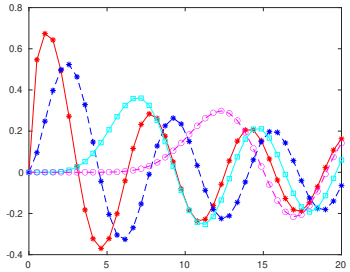

Figure 4: bessel fuction