SymForce

SYMFORCE EXPLORATION

ADLAN AFIF NUGROHO

1103194089

ADLAN AFIF NUGROHO
1103194089

https://github.com/Mrlumutz/robotic-projects

SYMFORCE

Symforce adalah komputasi simbolik cepat dan library code pada aplikasi robotic seperti computer vision, state estimation, Motion planning dan controls

BAHASA PEMOGRAMAN

C++

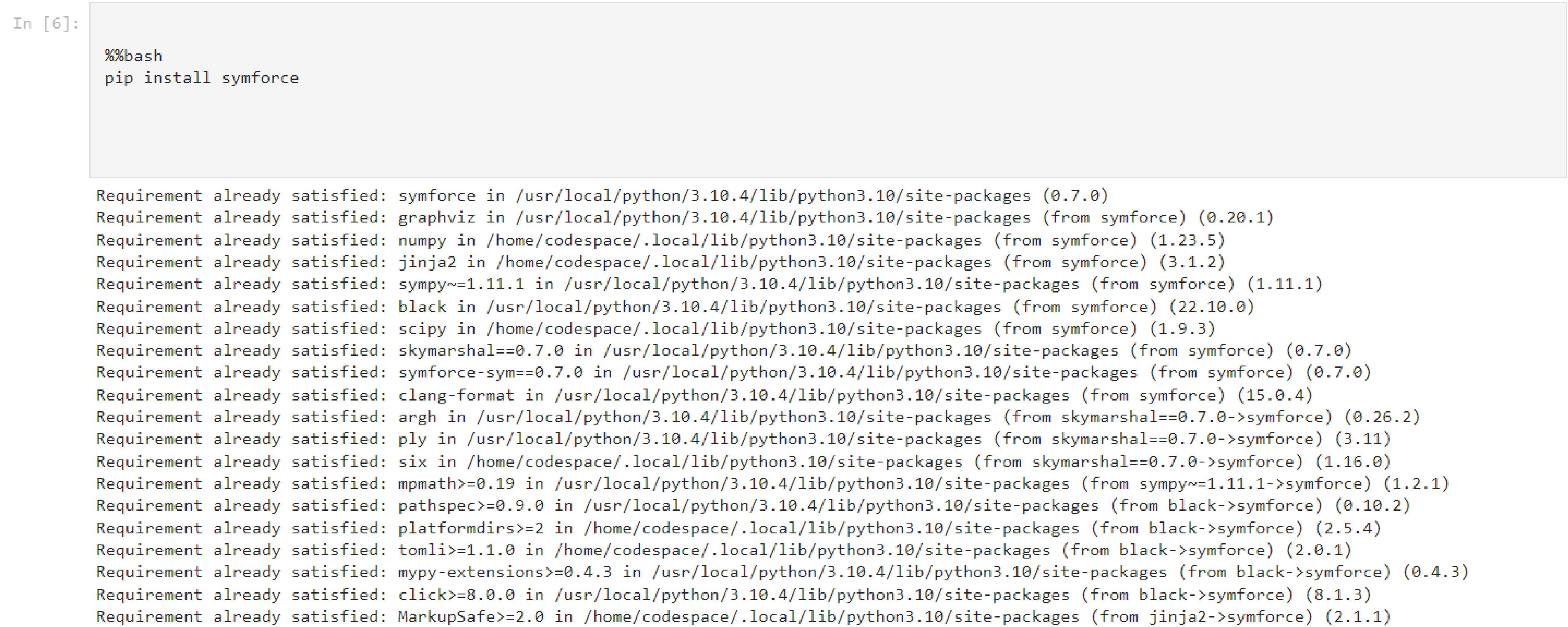
PYTHON

TUTORIAL INSTALLATION

Instalasi dan penggunaan symforce dapat diterapkan menggunakan VS Code atau juga di Codespaces yang ada di Github.

Untuk Step by Stepnya akan dilampirkan di slide selanjutnya

STEP 1

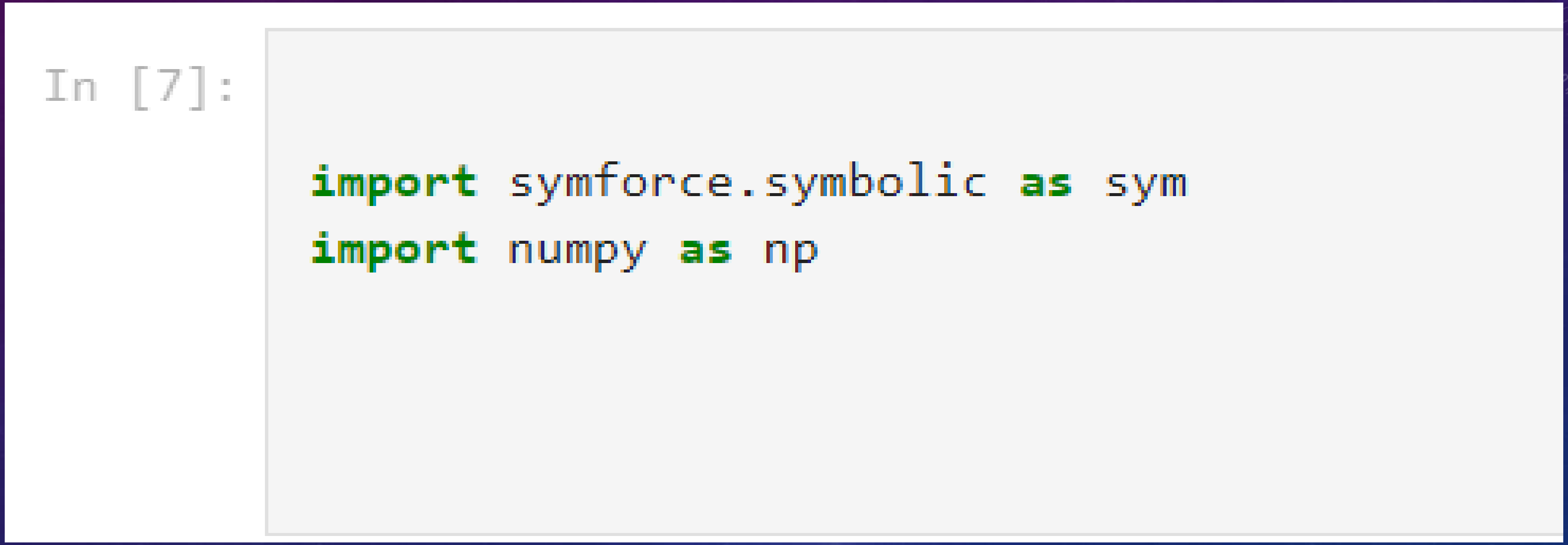


In [6]:

```
%%bash
pip install symforce
```

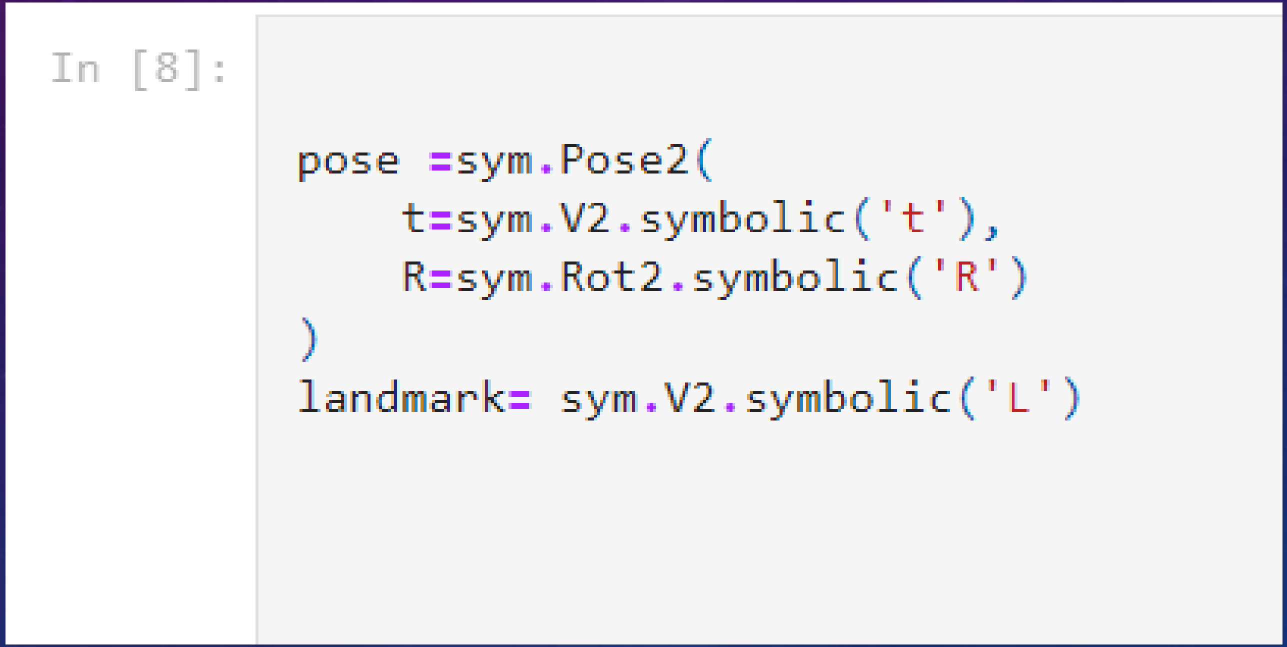
```
Requirement already satisfied: symforce in /usr/local/python/3.10.4/lib/python3.10/site-packages (0.7.0)
Requirement already satisfied: graphviz in /usr/local/python/3.10.4/lib/python3.10/site-packages (from symforce) (0.20.1)
Requirement already satisfied: numpy in /home/codespace/.local/lib/python3.10/site-packages (from symforce) (1.23.5)
Requirement already satisfied: jinja2 in /home/codespace/.local/lib/python3.10/site-packages (from symforce) (3.1.2)
Requirement already satisfied: sympy~=1.11.1 in /usr/local/python/3.10.4/lib/python3.10/site-packages (from symforce) (1.11.1)
Requirement already satisfied: black in /usr/local/python/3.10.4/lib/python3.10/site-packages (from symforce) (22.10.0)
Requirement already satisfied: scipy in /home/codespace/.local/lib/python3.10/site-packages (from symforce) (1.9.3)
Requirement already satisfied: skymarshal==0.7.0 in /usr/local/python/3.10.4/lib/python3.10/site-packages (from symforce) (0.7.0)
Requirement already satisfied: symforce-sym==0.7.0 in /usr/local/python/3.10.4/lib/python3.10/site-packages (from symforce) (0.7.0)
Requirement already satisfied: clang-format in /usr/local/python/3.10.4/lib/python3.10/site-packages (from symforce) (15.0.4)
Requirement already satisfied: argh in /usr/local/python/3.10.4/lib/python3.10/site-packages (from skymarshal==0.7.0->symforce) (0.26.2)
Requirement already satisfied: ply in /usr/local/python/3.10.4/lib/python3.10/site-packages (from skymarshal==0.7.0->symforce) (3.11)
Requirement already satisfied: six in /home/codespace/.local/lib/python3.10/site-packages (from skymarshal==0.7.0->symforce) (1.16.0)
Requirement already satisfied: mpmath>=0.19 in /usr/local/python/3.10.4/lib/python3.10/site-packages (from sympy~=1.11.1->symforce) (1.2.1)
Requirement already satisfied: pathspec>=0.9.0 in /usr/local/python/3.10.4/lib/python3.10/site-packages (from black->symforce) (0.10.2)
Requirement already satisfied: platformdirs>=2 in /home/codespace/.local/lib/python3.10/site-packages (from black->symforce) (2.5.4)
Requirement already satisfied: tomli>=1.1.0 in /home/codespace/.local/lib/python3.10/site-packages (from black->symforce) (2.0.1)
Requirement already satisfied: mypy-extensions>=0.4.3 in /usr/local/python/3.10.4/lib/python3.10/site-packages (from black->symforce) (0.4.3)
Requirement already satisfied: click>=8.0.0 in /usr/local/python/3.10.4/lib/python3.10/site-packages (from black->symforce) (8.1.3)
Requirement already satisfied: MarkupSafe>=2.0 in /home/codespace/.local/lib/python3.10/site-packages (from jinja2->symforce) (2.1.1)
```

STEP 2



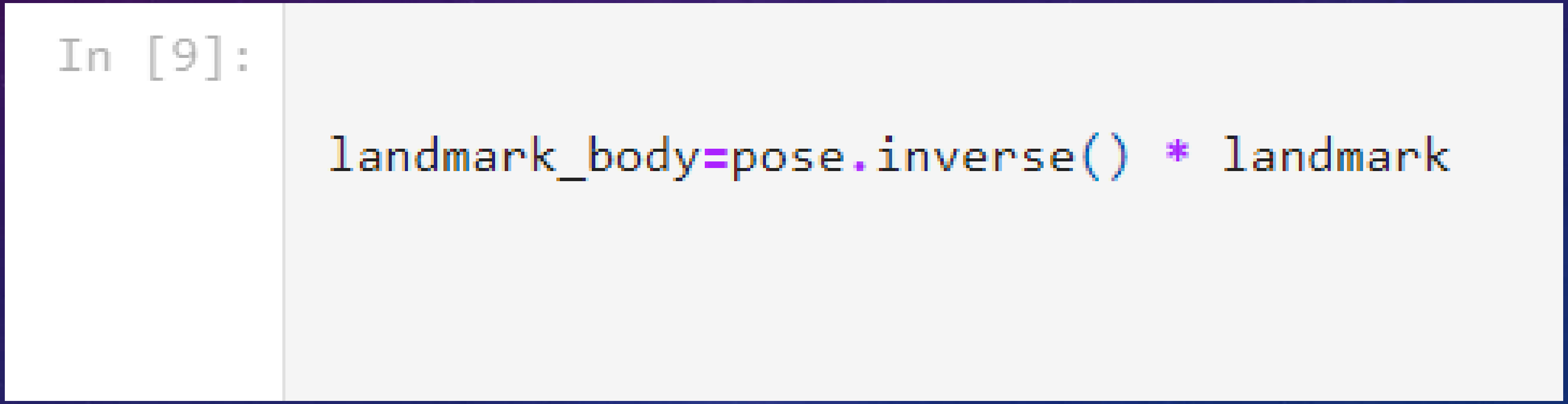
```python
import symforce.symbolic as sym
import numpy as np
```

STEP 3



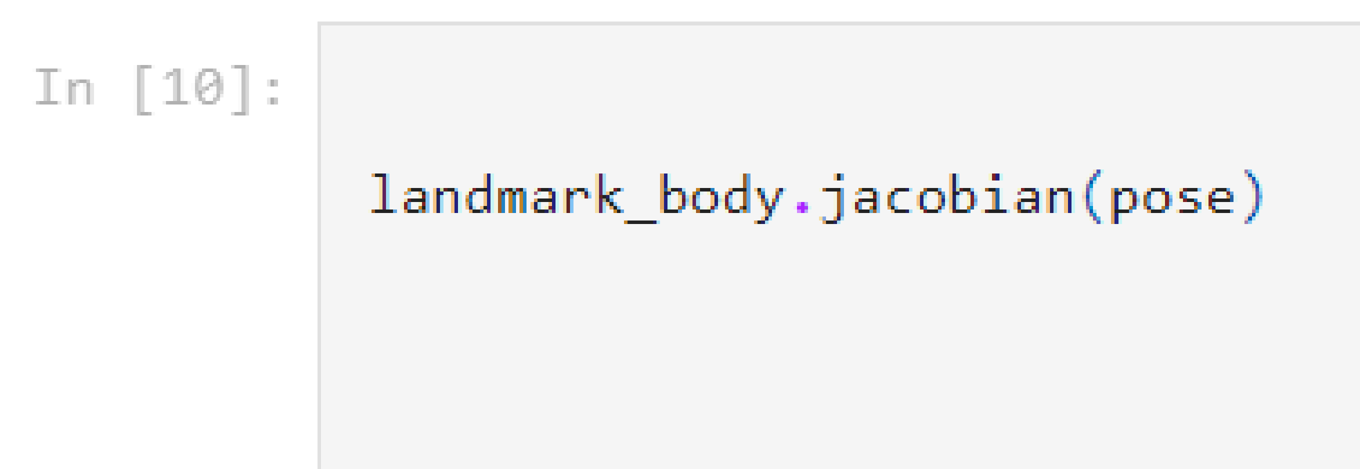
```
In [8]:

pose =sym.Pose2(
    t=sym.V2.symbolic('t'),
    R=sym.Rot2.symbolic('R')
)
landmark= sym.V2.symbolic('L')
```

STEP 4

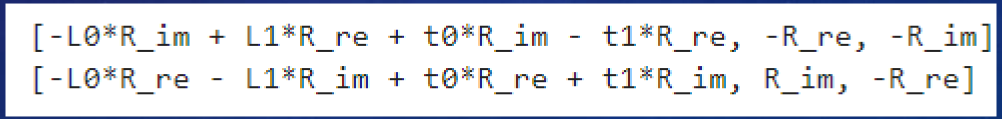


```
landmark_body=pose.inverse() * landmark
```

STEP 5

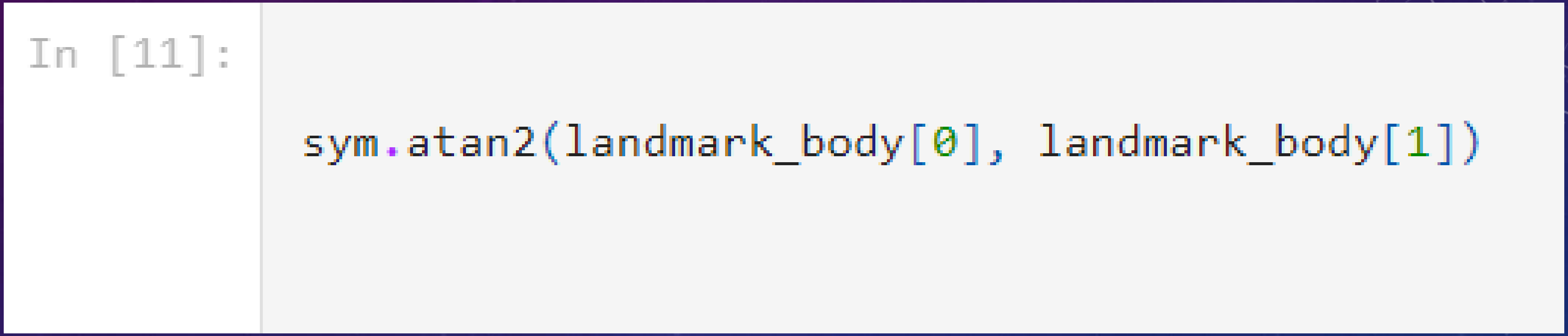


```
landmark_body.jacobian(pose)
```

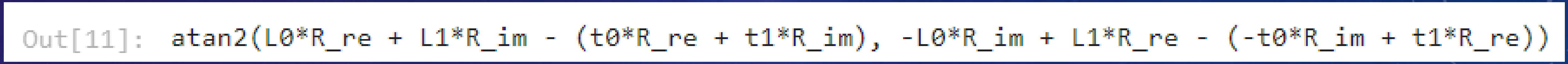


```
[-L0*R_im + L1*R_re + t0*R_im - t1*R_re, -R_re, -R_im]
[-L0*R_re - L1*R_im + t0*R_re + t1*R_im, R_im, -R_re]
```

STEP 6

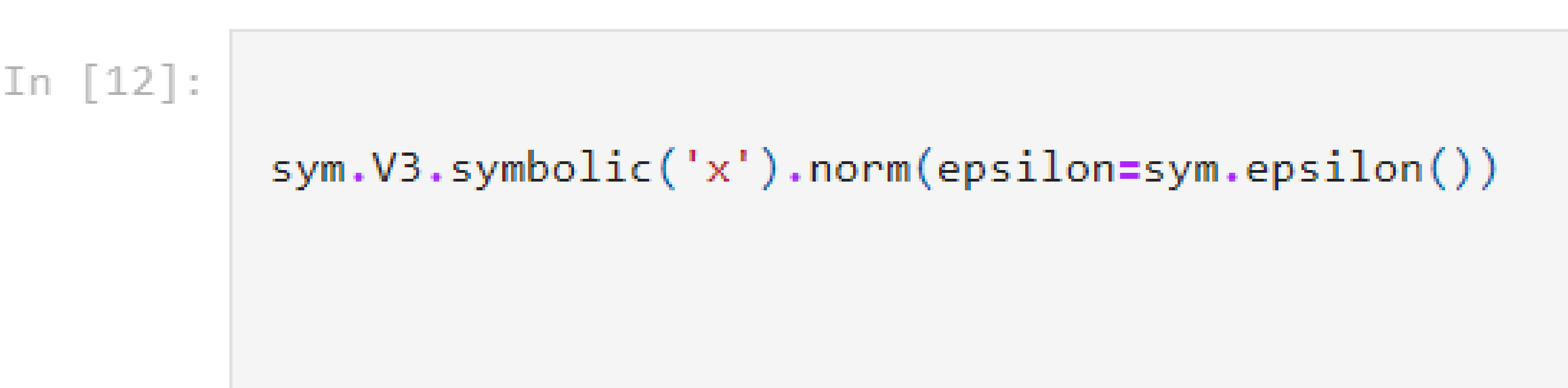


```
In [11]:  sym.atan2(landmark_body[0], landmark_body[1])
```

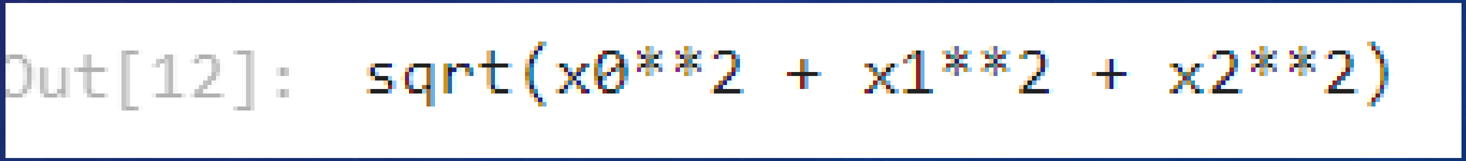


```
Out[11]:  atan2(L0*R_re + L1*R_im - (t0*R_re + t1*R_im), -L0*R_im + L1*R_re - (-t0*R_im + t1*R_re))
```

STEP 7

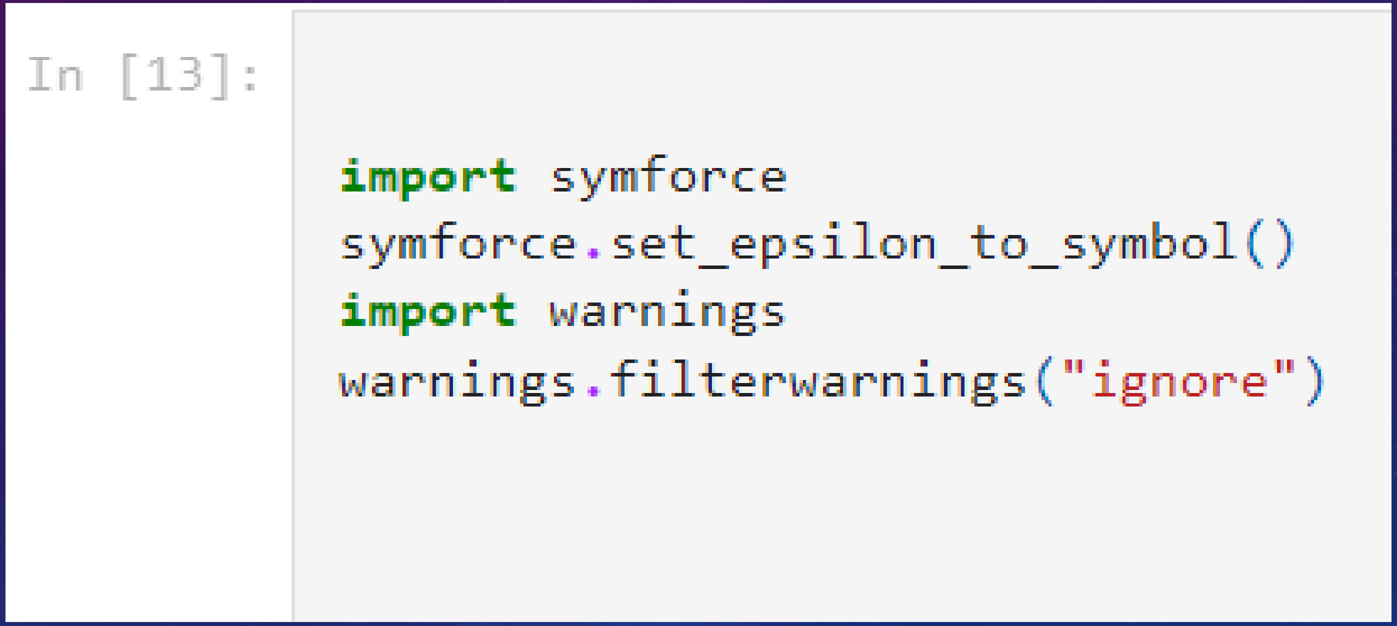


```
In [12]:  sym.V3.symbolic('x').norm(epsilon=sym.epsilon())
```



```
Out[12]:  sqrt(x0**2 + x1**2 + x2**2)
```

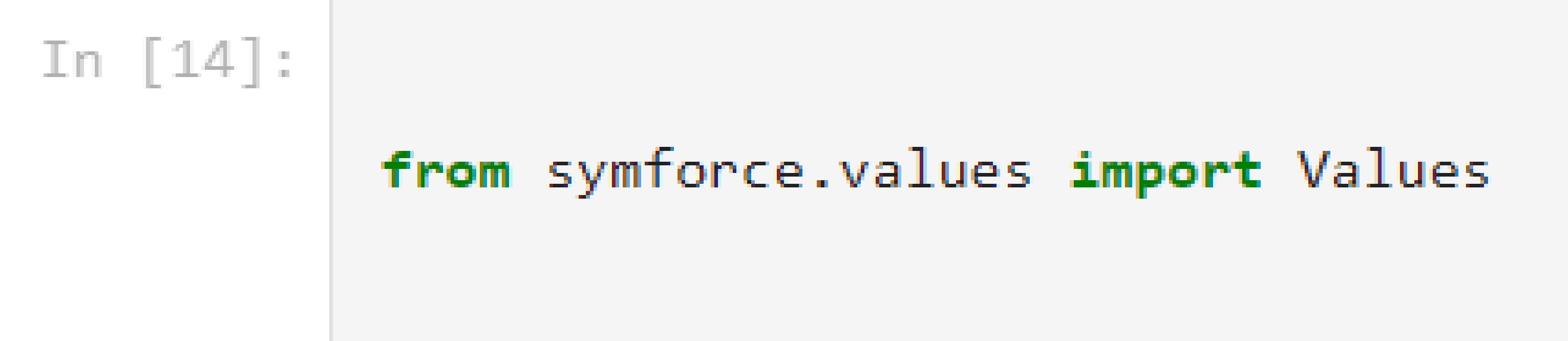
STEP 8



In [13]:

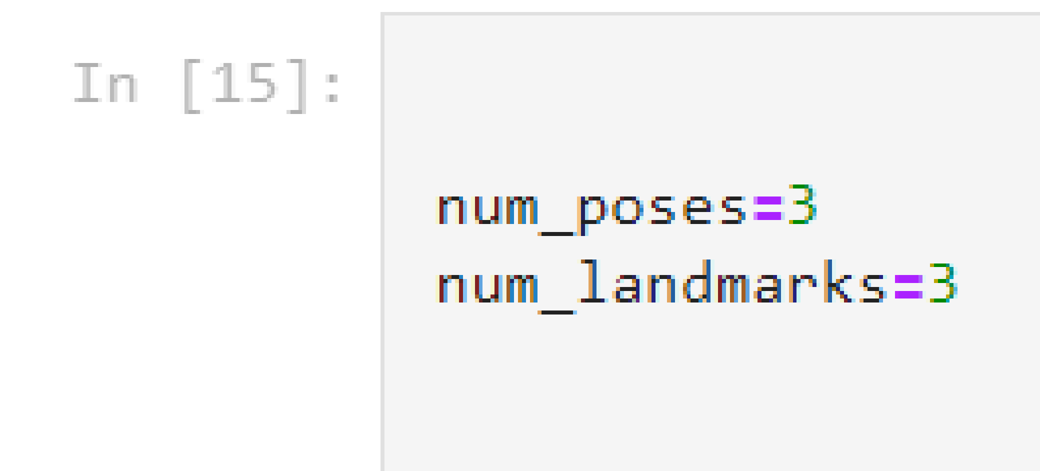
```
import symforce
symforce.set_epsilon_to_symbol()
import warnings
warnings.filterwarnings("ignore")
```

STEP 9



```
from symforce.values import Values
```

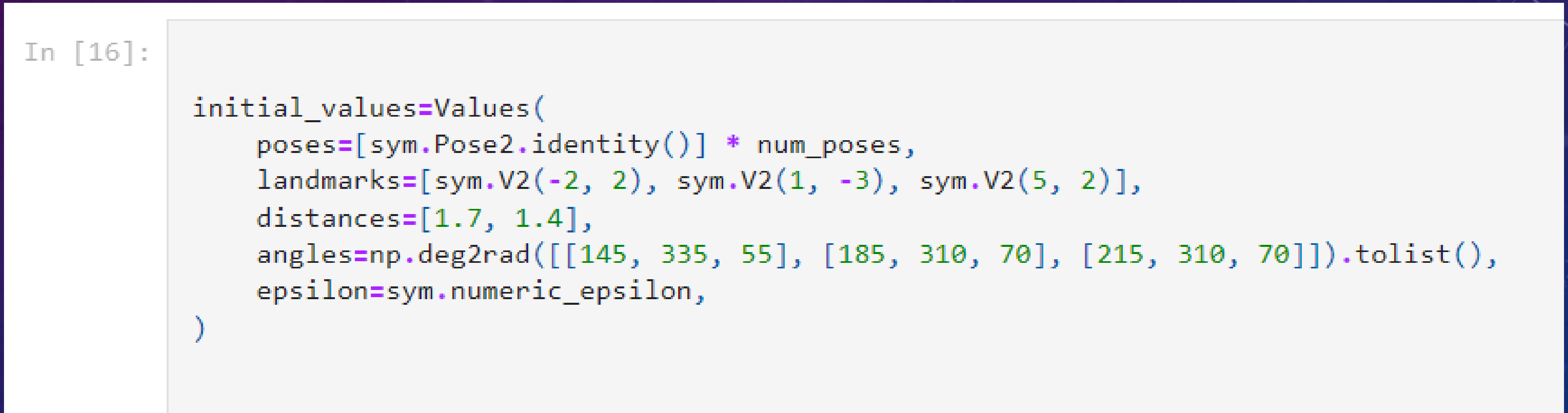
STEP 10



In [15]:

```
num_poses=3
num_landmarks=3
```

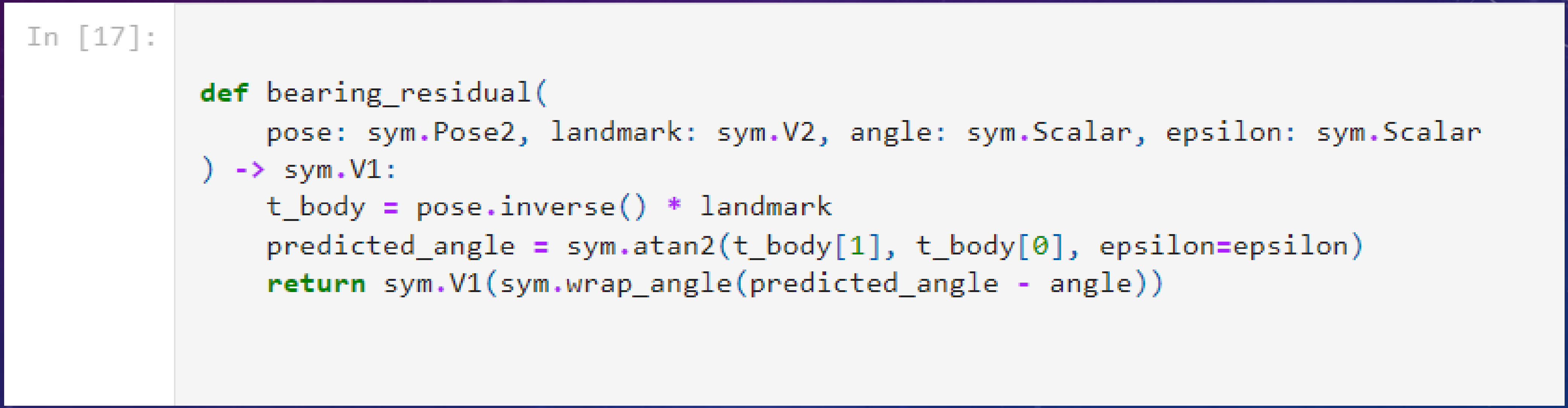
STEP 11



In [16]:

```
initial_values=Values(
    poses=[sym.Pose2.identity()] * num_poses,
    landmarks=[sym.V2(-2, 2), sym.V2(1, -3), sym.V2(5, 2)],
    distances=[1.7, 1.4],
    angles=np.deg2rad([[145, 335, 55], [185, 310, 70], [215, 310, 70]]).tolist(),
    epsilon=sym.numeric_epsilon,
)
```

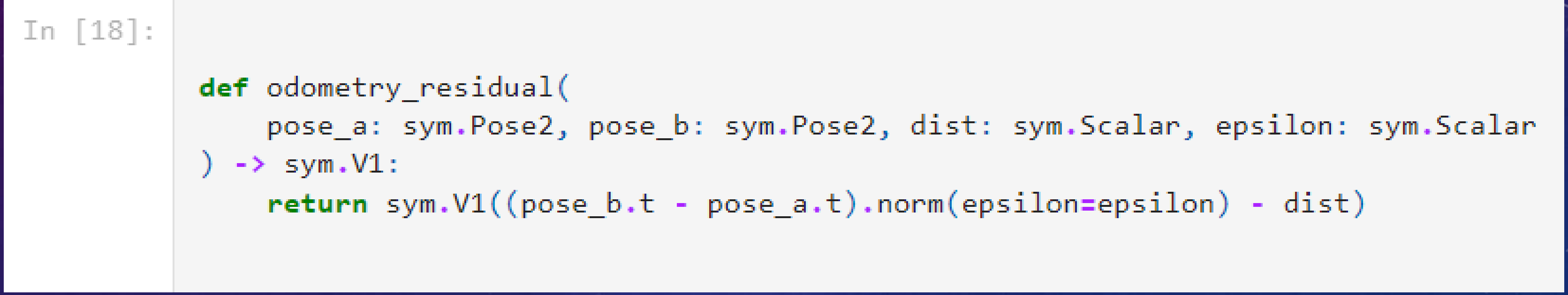
STEP 12



In [17]:

```
def bearing_residual(
    pose: sym.Pose2, landmark: sym.V2, angle: sym.Scalar, epsilon: sym.Scalar
) -> sym.V1:
    t_body = pose.inverse() * landmark
    predicted_angle = sym.atan2(t_body[1], t_body[0], epsilon=epsilon)
    return sym.V1(sym.wrap_angle(predicted_angle - angle))
```

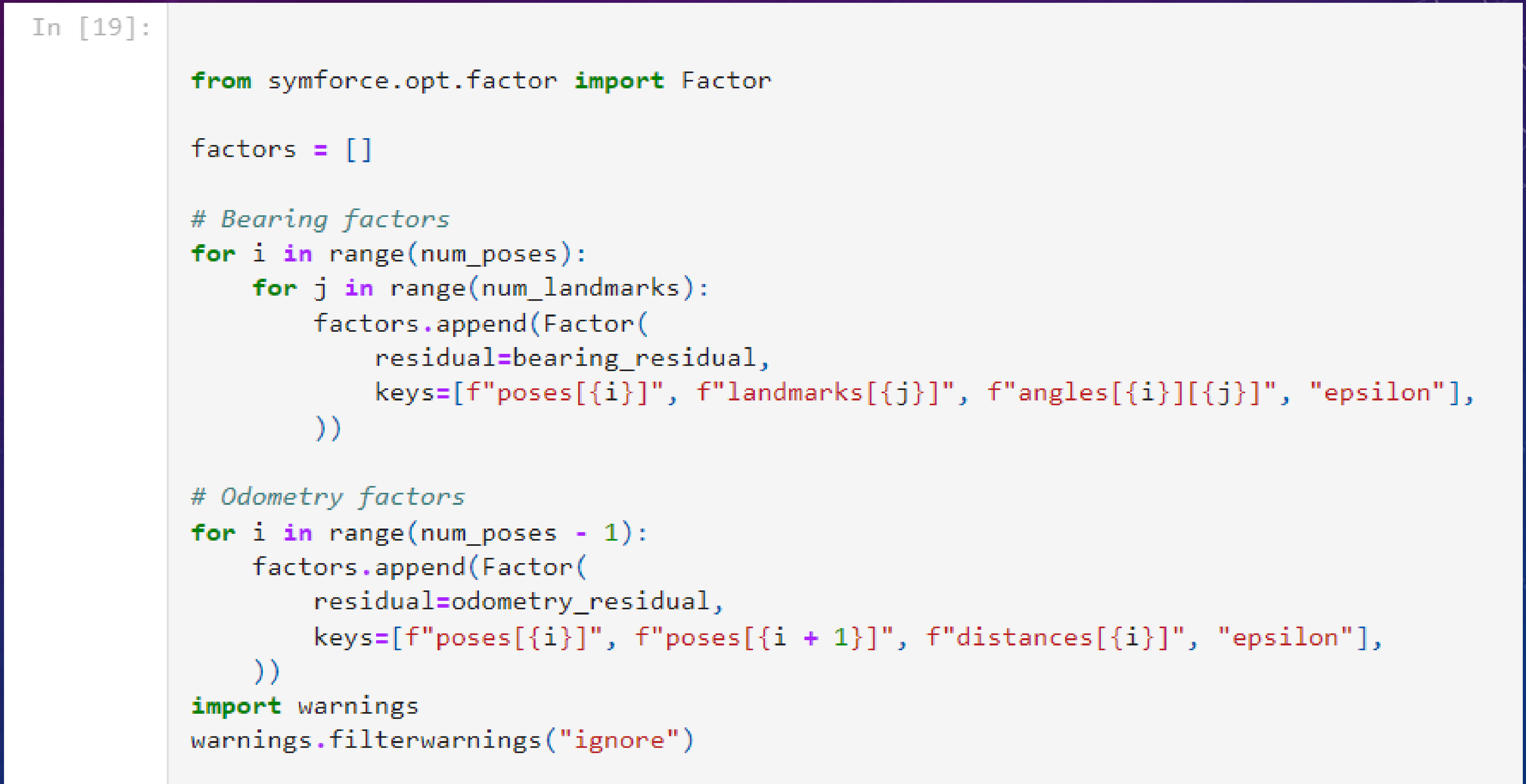
STEP 13



In [18]:

```python
def odometry_residual(
    pose_a: sym.Pose2, pose_b: sym.Pose2, dist: sym.Scalar, epsilon: sym.Scalar
) -> sym.V1:
    return sym.V1((pose_b.t - pose_a.t).norm(epsilon=epsilon) - dist)
```

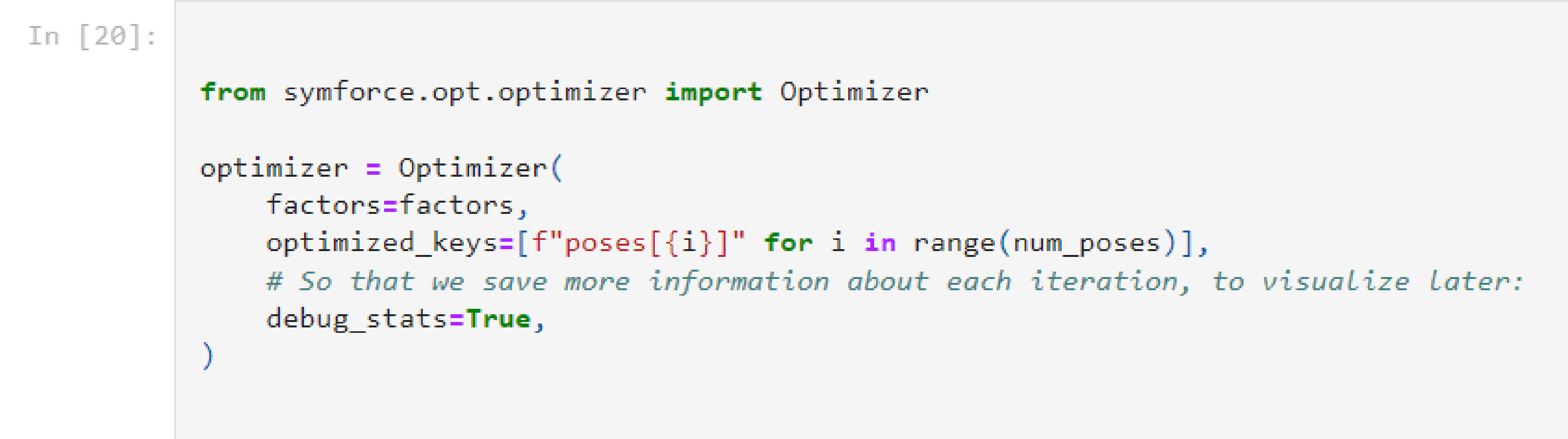
STEP 14



In [19]:

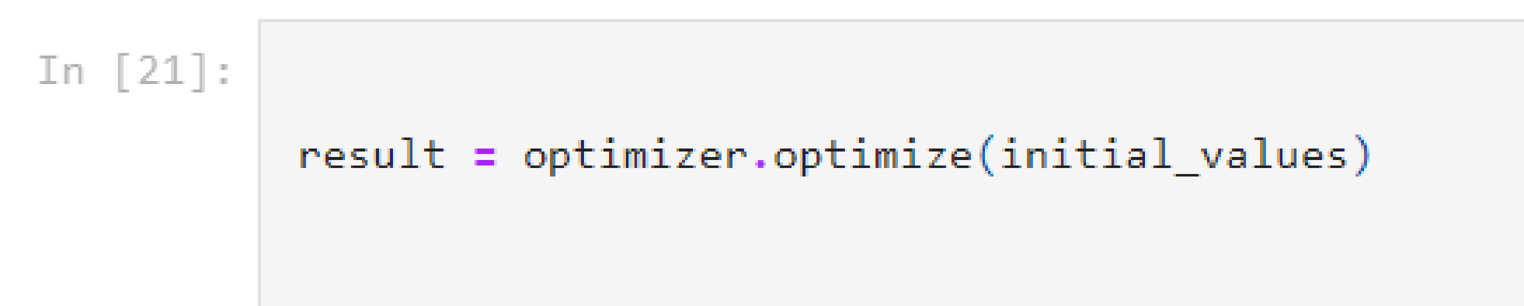
```python
from symforce.opt.factor import Factor

factors = []

# Bearing factors
for i in range(num_poses):
    for j in range(num_landmarks):
        factors.append(Factor(
            residual=bearing_residual,
            keys=[f"poses[{i}]", f"landmarks[{j}]", f"angles[{i}][{j}]", "epsilon"],
        ))

# Odometry factors
for i in range(num_poses - 1):
    factors.append(Factor(
        residual=odometry_residual,
        keys=[f"poses[{i}]", f"poses[{i + 1}]", f"distances[{i}]", "epsilon"],
    ))
import warnings
warnings.filterwarnings("ignore")
```

STEP 15



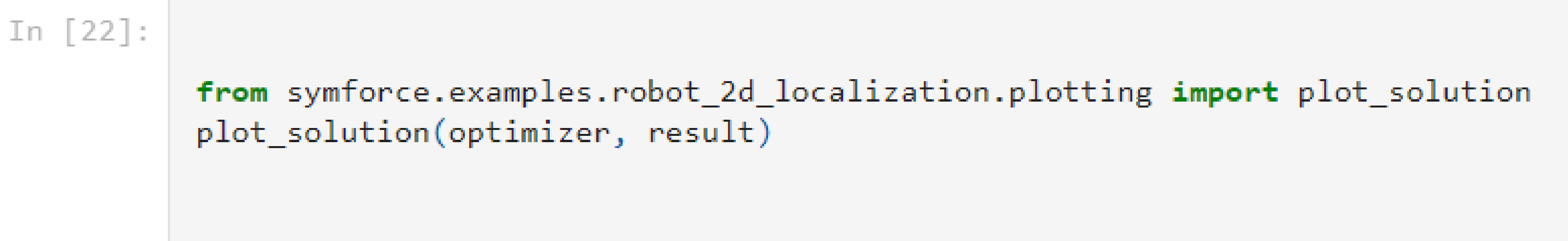
```python
from symforce.opt.optimizer import Optimizer

optimizer = Optimizer(
    factors=factors,
    optimized_keys=[f"poses[{i}]" for i in range(num_poses)],
    # So that we save more information about each iteration, to visualize later:
    debug_stats=True,
)
```

STEP 16



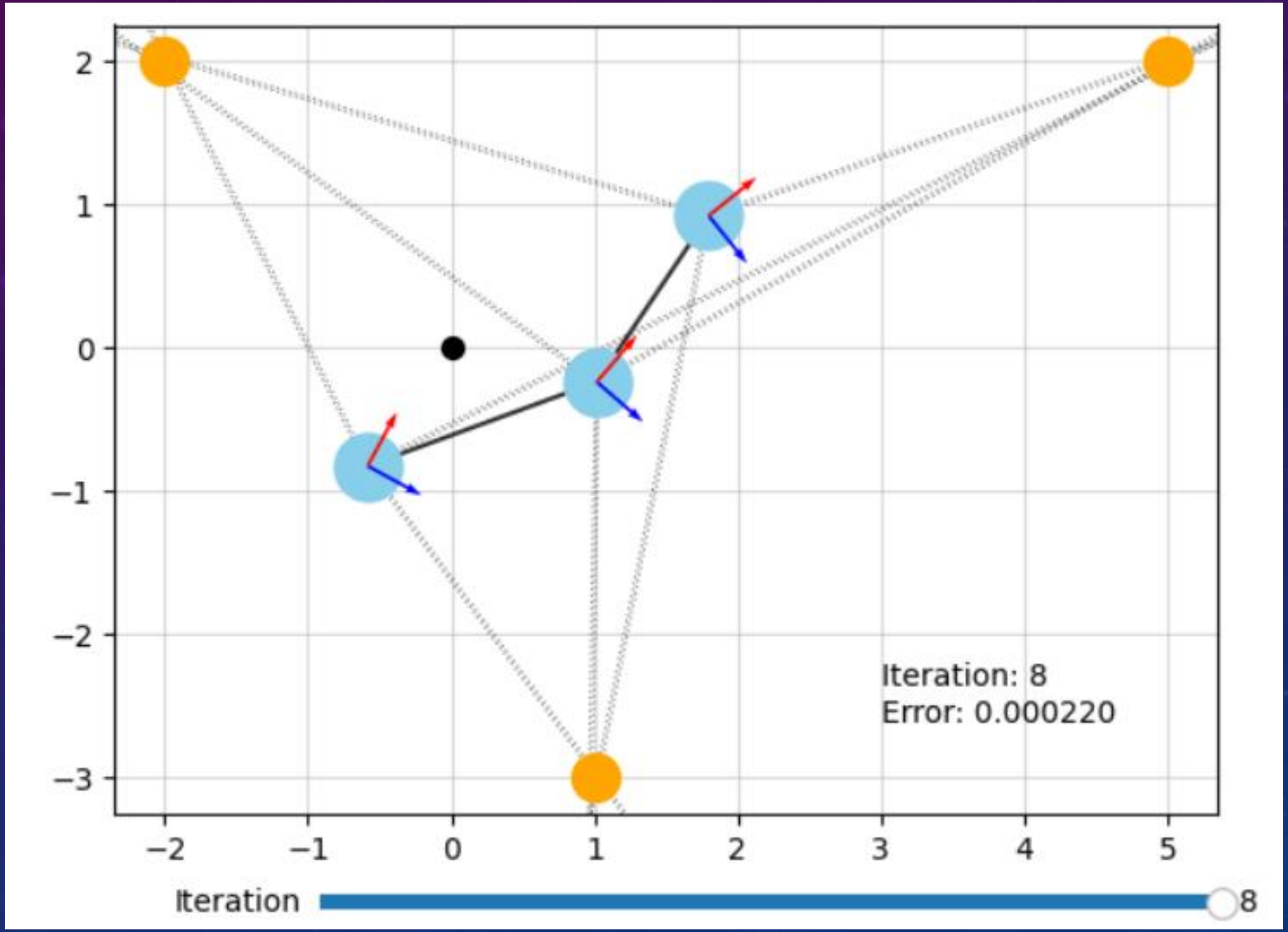
```
result = optimizer.optimize(initial_values)
```

STEP 17

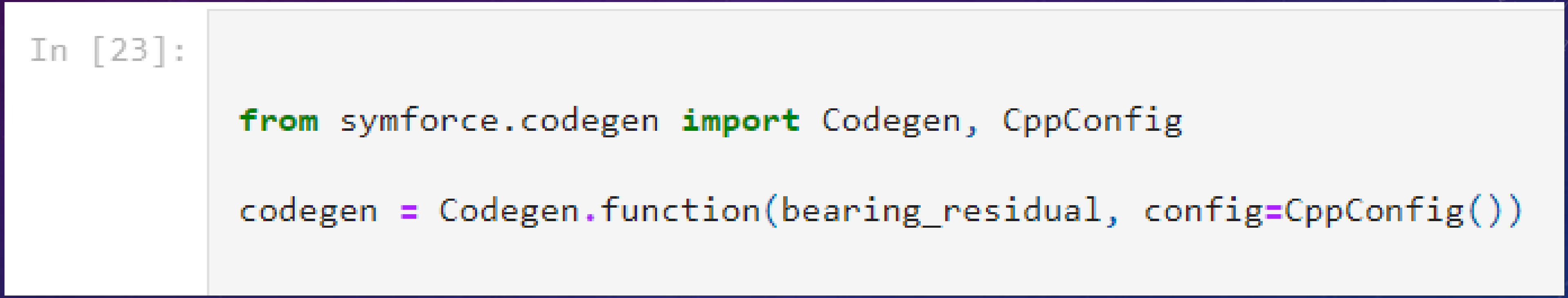


In [22]:

```
from symforce.examples.robot_2d_localization.plotting import plot_solution
plot_solution(optimizer, result)
```

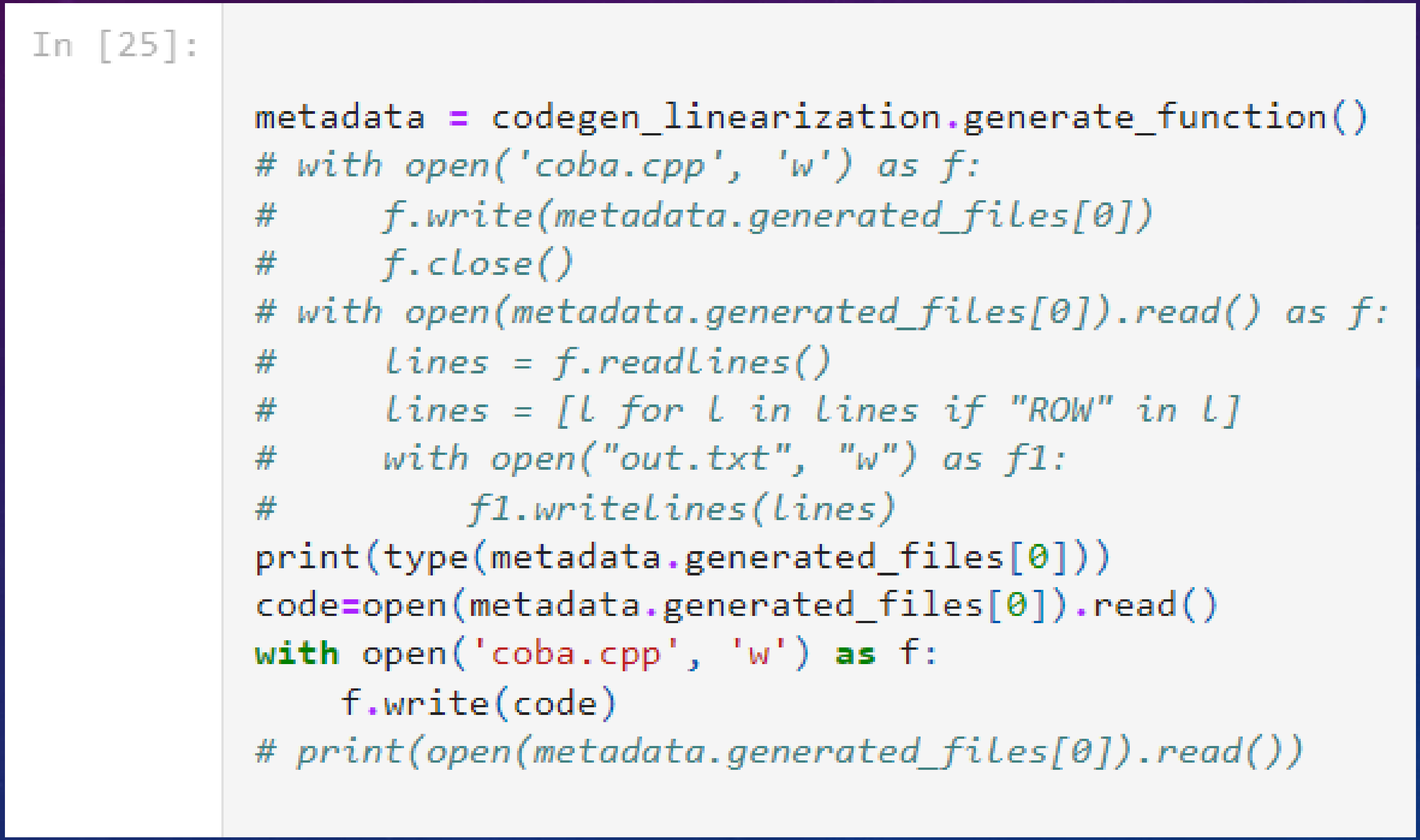

2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
Iteration: 8
Error: 0.000220
Iteration
8

STEP 18



```
In [23]:

from symforce.codegen import Codegen, CppConfig

codegen = Codegen.function(bearing_residual, config=CppConfig())
```

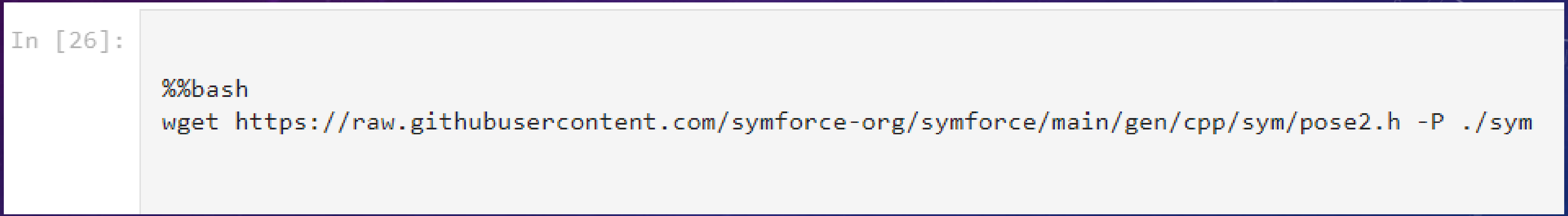
STEP 19



In [25]:

```
metadata = codegen_linearization.generate_function()
# with open('coba.cpp', 'w') as f:
#     f.write(metadata.generated_files[0])
#     f.close()
# with open(metadata.generated_files[0]).read() as f:
#     lines = f.readlines()
#     lines = [l for l in lines if "ROW" in l]
#     with open("out.txt", "w") as f1:
#         f1.writelines(lines)
print(type(metadata.generated_files[0]))
code=open(metadata.generated_files[0]).read()
with open('coba.cpp', 'w') as f:
    f.write(code)
# print(open(metadata.generated_files[0]).read())
```

STEP 20



```
%%bash
wget https://raw.githubusercontent.com/symforce-org/symforce/main/gen/cpp/sym/pose2.h -P ./sym
```