```
pip install symforce
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting symforce
  Downloading symforce-0.7.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.4 MB)
     |████████████████████████████████| 4.4 MB 5.4 MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from symforce) (1.21.6)
Collecting skymarshal==0.7.0
  Downloading skymarshal-0.7.0-py3-none-any.whl (82 kB)
     |████████████████████████████████| 82 kB 312 kB/s
Collecting sympy~=1.11.1
  Downloading sympy-1.11.1-py3-none-any.whl (6.5 MB)
     |████████████████████████████████| 6.5 MB 42.5 MB/s
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from symforce) (1.7.3)
Collecting clang-format
  Downloading clang_format-15.0.4-py2.py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 23.6 MB/s
Collecting black
  Downloading black-22.10.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 50.7 MB/s
Requirement already satisfied: jinja2 in /usr/local/lib/python3.8/dist-packages (from symforce) (2.11.3)
Collecting symforce-sym==0.7.0
  Downloading symforce_sym-0.7.0-py3-none-any.whl (70 kB)
     |████████████████████████████████| 70 kB 4.6 MB/s
Requirement already satisfied: graphviz in /usr/local/lib/python3.8/dist-packages (from symforce) (0.10.1)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages (from skymarshal==0.7.0->symforce) (1.15.0)
Collecting ply
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
     |████████████████████████████████| 49 kB 3.9 MB/s
Collecting argh
  Downloading argh-0.26.2-py2.py3-none-any.whl (30 kB)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.8/dist-packages (from sympy~=1.11.1->symforce) (1.2.1)
Requirement already satisfied: typing-extensions>=3.10.0.0 in /usr/local/lib/python3.8/dist-packages (from black->symforce) (4.1.1)
Requirement already satisfied: tomli>=1.1.0 in /usr/local/lib/python3.8/dist-packages (from black->symforce) (2.0.1)
Collecting platformdirs>=2
  Downloading platformdirs-2.5.4-py3-none-any.whl (14 kB)
Collecting pathspec>=0.9.0
  Downloading pathspec-0.10.2-py3-none-any.whl (28 kB)
Collecting click>=8.0.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
     |████████████████████████████████| 96 kB 3.9 MB/s
Collecting mypy-extensions>=0.4.3
  Downloading mypy_extensions-0.4.3-py2.py3-none-any.whl (4.5 kB)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.8/dist-packages (from jinja2->symforce) (2.0.1)
Installing collected packages: ply, platformdirs, pathspec, mypy-extensions, click, argh, sympy, symforce-sym, skymarshal, clang-format,
  Attempting uninstall: click
    Found existing installation: click 7.1.2
    Uninstalling click-7.1.2:
      Successfully uninstalled click-7.1.2
  Attempting uninstall: sympy
    Found existing installation: sympy 1.7.1
    Uninstalling sympy-1.7.1:
      Successfully uninstalled sympy-1.7.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
flask 1.1.4 requires click<8.0,>=5.1, but you have click 8.1.3 which is incompatible.
Successfully installed argh-0.26.2 black-22.10.0 clang-format-15.0.4 click-8.1.3 mypy-extensions-0.4.3 pathspec-0.10.2 platformdirs-2.5
```

```
import numpy as np
import os
import symforce

symforce.set_symbolic_api("symengine")
symforce.set_log_level("warning")

# https://symforce.org/tutorials/epsilon_tutorial.html
symforce.set_epsilon_to_symbol()

from symforce import codegen
from symforce.codegen import codegen_util
from symforce import ops
import symforce.symbolic as sf
```

```
inputs.add(sf.Symbol("foo"))
display(inputs)
```

```
Values(
  x: x,
  y: <Rot2 <C real=c_re, imag=c_im>>,
  foo: foo,
)
```

```
x, y = sf.symbols("x y")
expr = x ** 2 + sf.sin(y) / x ** 2
inputs["states"] = Values(p=expr)
display(inputs)
```

```
Values(
  x: x,
  y: <Rot2 <C real=c_re, imag=c_im>>,
  foo: foo,
  states:   Values(
    p: x**2 + sin(y)/x**2,
  ),
)
```

```
display(inputs.to_storage())
```

$$\left[x,\ c_{re},\ c_{im},\ foo,\ x^{2} + \frac{\sin\left(y\right)}{x^{2}}\right]$$

```
display(inputs.items_recursive())
```

```
[('x', x),
 ('y', <Rot2 <C real=c_re, imag=c_im>>),
 ('foo', foo),
 ('states.p', x**2 + sin(y)/x**2)]
```

```
display(inputs.keys_recursive())
display(inputs.values_recursive())
```

```
['x', 'y', 'foo', 'states.p']
[x, <Rot2 <C real=c_re, imag=c_im>>, foo, x**2 + sin(y)/x**2]
```

```
index = inputs.index()
index
```

```
OrderedDict([('x',
  IndexEntry(offset=0, storage_dim=1, _module='builtins', _qualname='float', shape=None, item_index=None)),
 ('y',
  IndexEntry(offset=1, storage_dim=2, _module='symforce.geo.rot2', _qualname='Rot2', shape=None, item_index=None)),
 ('foo',
  IndexEntry(offset=3, storage_dim=1, _module='builtins', _qualname='float', shape=None, item_index=None)),
 ('states',
  IndexEntry(offset=4, storage_dim=1, _module='symforce.values.values', _qualname='Values', shape=None, item_index=OrderedDict([('p',
IndexEntry(offset=0, storage_dim=1, _module='builtins', _qualname='float', shape=None, item_index=None))]))])])
```

```
inputs2 = Values.from_storage_index(inputs.to_storage(), index)
assert inputs == inputs2
display(inputs)
```

```
Values(
  x: x,
  y: <Rot2 <C real=c_re, imag=c_im>>,
  foo: foo,
  states:   Values(
    p: x**2 + sin(y)/x**2,
  ),
)
```

```
item_index = inputs.index()["states"].item_index
assert item_index == inputs["states"].index()
```

```
inputs["states.blah"] = 3
display(inputs)
```

```
Values(
  x: x,
  y: <Rot2 <C real=c_re, imag=c_im>>,
  foo: foo,
  states:   Values(
    p: x**2 + sin(y)/x**2,
    blah: 3,
  ),
)
```

```
assert inputs["states.p"] is inputs["states"]["p"] is inputs.attr.states.p
display(inputs.attr.states.p)
```

$$x^{2} + \frac{\sin\left(y\right)}{x^{2}}$$

```
with sf.scope("params"):
    s = sf.Symbol("cost")
display(s)
```

$$params.cost$$

```
v = Values()
v.add(sf.Symbol("x"))
with sf.scope("foo"):
    v.add(sf.Symbol("x"))
    with sf.scope("bar"):
        v.add(sf.Symbol("x"))
display(v)
display(v.attr.foo.bar.x)
```

```
Values(
  x: x,
  foo:   Values(
    x: foo.x,
    bar:     Values(
      x: foo.bar.x,
    ),
  ),
)
```

$$foo.bar.x$$

```
v = Values()
with v.scope("hello"):
    v["y"] = x ** 2
    v["z"] = sf.Symbol("z")
v
```

```
 Values(
   hello:   Values(
     y: x**2,
     z: hello.z,
   ),
 )
```

```
lie_vals = Values()
lie_vals["scalar"] = sf.Symbol("x")
lie_vals["rot3"] = sf.Rot3.symbolic("rot")

sub_lie_vals = Values()
sub_lie_vals["pose3"] = sf.Pose3.symbolic("pose")
sub_lie_vals["vec"] = sf.V3.symbolic("vec")

lie_vals["sub_vals"] = sub_lie_vals

display(lie_vals)
```

```
Values(
  scalar: x,
  rot3: <Rot3 <Q xyzw=[rot_x, rot_y, rot_z, rot_w]>>,
  sub_vals:   Values(
    pose3: <Pose3 R=<Rot3 <Q xyzw=[pose.R_x, pose.R_y, pose.R_z, pose.R_w]>>, t=(pose.t0, pose.t1, pose.t2)>,
    vec: Matrix([
[vec0],
[vec1],
[vec2]]),
```

```
display(lie_vals.tangent_dim())
display(len(lie_vals.to_tangent()))
```

13
13

```
display(lie_vals.storage_D_tangent())
```

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{rot_w}{2} & -\frac{rot_z}{2} & \frac{rot_y}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{rot_z}{2} & \frac{rot_w}{2} & -\frac{rot_x}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{rot_y}{2} & \frac{rot_x}{2} & \frac{rot_w}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{rot_x}{2} & -\frac{rot_y}{2} & -\frac{rot_z}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{pose.R_w}{2} & -\frac{pose.R_z}{2} & \frac{pose.R_y}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{pose.R_z}{2} & \frac{pose.R_w}{2} & -\frac{pose.R_x}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{pose.R_y}{2} & \frac{pose.R_x}{2} & \frac{pose.R_w}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{pose.R_x}{2} & -\frac{pose.R_y}{2} & -\frac{pose.R_z}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

```
residual = sf.Matrix(6, 1)
residual[0:3, 0] = lie_vals["rot3"] * lie_vals["sub_vals.vec"]
residual[3:6, 0] = lie_vals["sub_vals.pose3"] * lie_vals["sub_vals.vec"]
display(residual)
```

$$\begin{bmatrix} vec_0\left(-2rot_y^2 - 2rot_z^2 + 1\right) + vec_1\left(-2rot_w rot_z + 2rot_x rot_y\right) + vec_2 \cdot \left(2rot_w rot_y + 2rot_x rot_z\right) \\ vec_0 \cdot \left(2rot_w rot_z + 2rot_x rot_y\right) + vec_1\left(-2rot_x^2 - 2rot_z^2 + 1\right) + vec_2\left(-2rot_w rot_x + 2rot_y rot_z\right) \\ vec_0\left(-2rot_w rot_y + 2rot_x rot_z\right) + vec_1 \cdot \left(2rot_w rot_x + 2rot_y rot_z\right) + vec_2\left(-2rot_x^2 - 2rot_y^2 + 1\right) \\ pose.t0 + vec_0\left(-2pose.R_y^2 - 2pose.R_z^2 + 1\right) + vec_1\left(-2pose.R_w pose.R_z + 2pose.R_x pose.R_y\right) + vec_2 \cdot \left(2pose.R_w pose.R_y + 2pose.R \\ pose.t1 + vec_0 \cdot \left(2pose.R_w pose.R_z + 2pose.R_x pose.R_y\right) + vec_1\left(-2pose.R_x^2 - 2pose.R_z^2 + 1\right) + vec_2\left(-2pose.R_w pose.R_x + 2pose.R \\ pose.t2 + vec_0\left(-2pose.R_w pose.R_y + 2pose.R_x pose.R_z\right) + vec_1 \cdot \left(2pose.R_w pose.R_x + 2pose.R_y pose.R_z\right) + vec_2\left(-2pose.R_x^2 - 2po \end{bmatrix}$$

```
residual_D_tangent = residual.jacobian(lie_vals)
display(residual_D_tangent.shape)
display(residual_D_tangent)
```

$$(6,\ 13)$$

$$\begin{bmatrix} 0 & vec_1 \cdot \left(2rot_w rot_y + 2rot_x rot_z\right) + vec_2 \cdot \left(2rot_w rot_z - 2rot_x rot_y\right) & vec_0\left(-2rot_w rot_y - 2rot_x rot_z\right) + vec_2\left(rot_w^2 + rot_x^2 - ro \\ 0 & vec_1\left(-2rot_w rot_x + 2rot_y rot_z\right) + vec_2\left(-rot_w^2 + rot_x^2 - rot_y^2 + rot_z^2\right) & vec_0 \cdot \left(2rot_w rot_x - 2rot_y rot_z\right) + vec_2 \cdot \left(2rot_w rot_z + 2r \\ 0 & vec_1\left(rot_w^2 - rot_x^2 - rot_y^2 + rot_z^2\right) + vec_2\left(-2rot_w rot_x - 2rot_y rot_z\right) & vec_0\left(-rot_w^2 + rot_x^2 + rot_y^2 - rot_z^2\right) + vec_2\left(-2rot_w rot_y + \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$