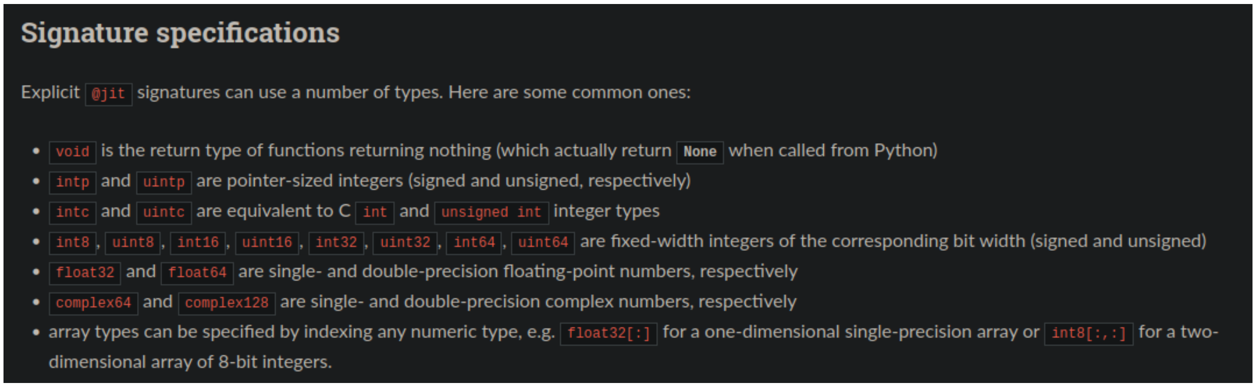
Use @jitclass, creating dictionaries or vector spaces that map the attributes to the intended data types

https://numba.pydata.org/numba-doc/latest/reference/types.html

Classes to convert to jitclasses:

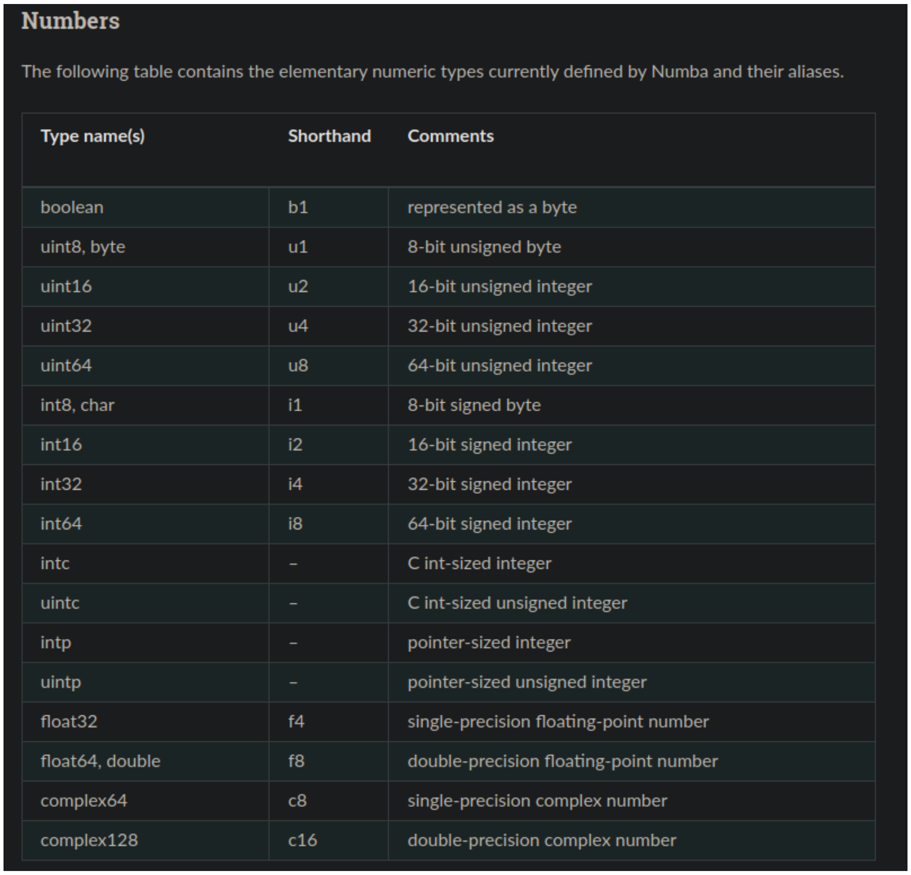
stochpy.implementations.StochPyTools.StochPySSA_Shared
stochpy.implementations.DirectMethod.DirectMethod
stochpy.modules.StochSim.SSASettings



Signature specifications

Explicit `@jit` signatures can use a number of types. Here are some common ones:

- `void` is the return type of functions returning nothing (which actually return `None` when called from Python)
- `intp` and `uintp` are pointer-sized integers (signed and unsigned, respectively)
- `intc` and `uintc` are equivalent to C `int` and `unsigned int` integer types
- `int8`, `uint8`, `int16`, `uint16`, `int32`, `uint32`, `int64`, `uint64` are fixed-width integers of the corresponding bit width (signed and unsigned)
- `float32` and `float64` are single- and double-precision floating-point numbers, respectively
- `complex64` and `complex128` are single- and double-precision complex numbers, respectively
- array types can be specified by indexing any numeric type, e.g. `float32[:]` for a one-dimensional single-precision array or `int8[:,:]` for a two-dimensional array of 8-bit integers.



Numbers

The following table contains the elementary numeric types currently defined by Numba and their aliases.

Type name(s)	Shorthand	Comments
boolean	b1	represented as a byte
uint8, byte	u1	8-bit unsigned byte
uint16	u2	16-bit unsigned integer
uint32	u4	32-bit unsigned integer
uint64	u8	64-bit unsigned integer
int8, char	i1	8-bit signed byte
int16	i2	16-bit signed integer
int32	i4	32-bit signed integer
int64	i8	64-bit signed integer
intc	–	C int-sized integer
uintc	–	C int-sized unsigned integer
intp	–	pointer-sized integer
uintp	–	pointer-sized unsigned integer
float32	f4	single-precision floating-point number
float64, double	f8	double-precision floating-point number
complex64	c8	single-precision complex number
complex128	c16	double-precision complex number

Numba documentation
https://numba.readthedocs.io/en/stable/

Python classes have,
Attributes
Methods

https://stackoverflow.com/questions/37768647/python-numba-fingerprint-error
https://www.google.com/search?channel=fs&client=ubuntu&q=python+numba+cannot+compute+fingerprint+of+empty+list

Don’t want to use empty lists

Supported python types
http://numba.pydata.org/numba-doc/0.26.0/reference/pysupported.html

Supported numpy types
http://numba.pydata.org/numba-doc/0.26.0/reference/numpysupported.html

Line 773, computationally expensive:

```
self.SSA.Execute(self.settings,IsStatusBar)
```

References:
stochpy/implementations/FastSingleMoleculeMethod.py

- argument 0: Cannot determine Numba type of <class 'stochpy.implementations.DirectMethod.DirectMethod'>

stochpy.implementations.DirectMethod.DirectMethod to numpy object

- argument 1: Cannot determine Numba type of <class 'stochpy.modules.StochSim.SSASettings'>

stochpy.modules.StochSim.SSASettings to numpy object