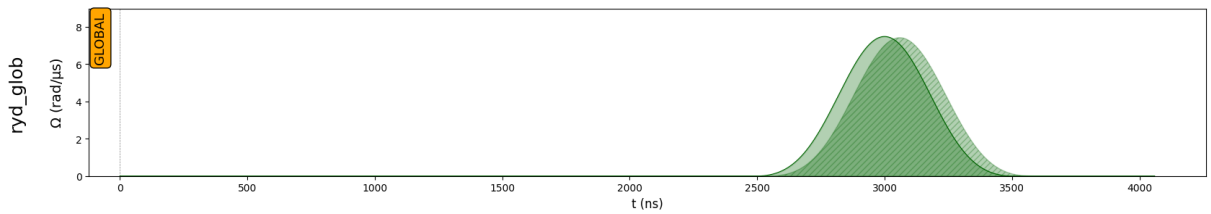
In [ ]:
```
import numpy as np
import pulser
import pulser_simulation
```

False positive simulations

The simulator outputs an incorrect result on develop

In [ ]:
```
seq = pulser.Sequence(pulser.Register.square(2, 5), pulser.AnalogDevice)
seq.declare_channel("ryd_glob", "rydberg_global")
seq.add(
    pulser.Pulse.ConstantDetuning(
        pulser.CompositeWaveform(
            pulser.ConstantWaveform(2500, 0.0),
            pulser.BlackmanWaveform(1000, np.pi),
            pulser.ConstantWaveform(500, 0.0),
            ),
        0,
        0,
        ),
    channel="ryd_glob",
)
seq.draw()
sim = pulser_simulation.QutipEmulator.from_sequence(seq)
```



Simulation on Pulser

In [ ]:
```
sim.run().get_final_state()
```

Out[ ]: Quantum object: dims = [[2, 2, 2, 2], [1, 1, 1, 1]], shape = (16, 1), type = ket

$$\begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \end{pmatrix}$$

This is now fixed

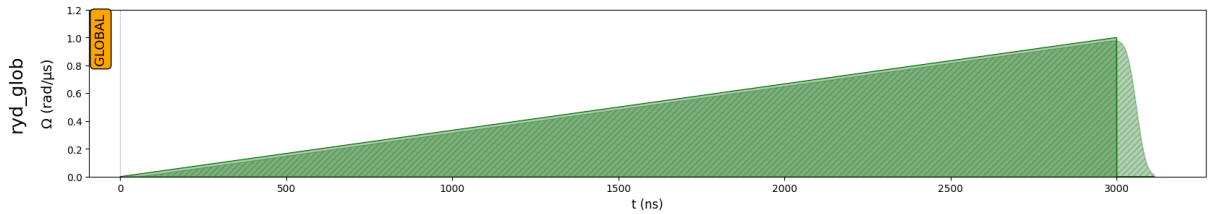
In [ ]:
```
sim.run().get_final_state()
```

Out[ ]: Quantum object: dims = [[2, 2, 2, 2], [1, 1, 1, 1]], shape = (16, 1), type = ket

$$\begin{pmatrix} (-1.915 \times 10^{-10} - 1.202 \times 10^{-10} j) \\ (7.526 \times 10^{-06} - 2.717 \times 10^{-06} j) \\ (7.526 \times 10^{-06} - 2.717 \times 10^{-06} j) \\ (5.337 \times 10^{-05} - 2.111 \times 10^{-06} j) \\ (7.526 \times 10^{-06} - 2.717 \times 10^{-06} j) \\ \vdots \\ (-0.147 + 0.104 j) \\ (5.337 \times 10^{-05} - 2.111 \times 10^{-06} j) \\ (-0.147 + 0.104 j) \\ (-0.147 + 0.104 j) \\ 0.695 \end{pmatrix}$$

A simulation with a high max_step on develop and high step size

In [ ]:
```
seq = pulser.Sequence(pulser.Register.square(2, 5), pulser.AnalogDevice)
seq.declare_channel("ryd_glob", "rydberg_global")
seq.add(
    pulser.Pulse.ConstantDetuning(
        pulser.RampWaveform(3000, 0, 1),
        0,
        0,
        ),
    channel="ryd_glob",
)
seq.draw()
sim = pulser_simulation.QutipEmulator.from_sequence(seq)
```



In [ ]:
```
%%timeit
# On develop
sim.run().get_final_state()
```

```
153 ms ± 1.75 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

In [ ]:
```
%%timeit
# On develop
sim.set_evaluation_times("Minimal")
sim.run().get_final_state()
```

```
11.4 ms ± 452 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

In [ ]:
```
%%timeit
# On current branch
sim.run().get_final_state()
```

```
140 ms ± 2.44 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

In [ ]:
```
%%timeit
# On current branch
sim.set_evaluation_times("Minimal")
sim.run().get_final_state()
```

```
18.2 ms ± 278 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```