time-reinitialization

February 19, 2026

[30]:

```
from openmm.app import *
from openmm import *
from openmm.unit import *
from sys import stdout
import random

pdb = PDBFile('liquid.pdb')
forcefield = ForceField('amber14/tip3pfb.xml')
system = forcefield.createSystem(pdb.topology, nonbondedMethod=PME,
        nonbondedCutoff=0.9*nanometer, constraints=HBonds)
integrator = LangevinMiddleIntegrator(300*kelvin, 1/picosecond, 0.
 ↪004*picoseconds)
simulation = Simulation(pdb.topology, system, integrator)
simulation.context.setPositions(pdb.positions)
```

[31]:

```
def update_parameters():
    for i in range(simulation.system.getNumForces()):
        if hasattr(simulation.system.getForce(i),'updateParametersInContext'):
            simulation.system.getForce(i).updateParametersInContext(simulation.
 ↪context)

def reinitialize_context():
    simulation.context.reinitialize(preserveState=True)

def randomize_parameters_and_update(func):
    for fc in simulation.system.getForces():
        if "Nonbond" in type(fc).__name__:
            units = [x.unit for x in fc.getParticleParameters(0)]
            for i in range(fc.getNumParticles()):
                q_ = random.random()
                sigma_ = random.random()
                eps_ = random.random()
                fc.setParticleParameters(i, q_ * units[0], sigma_ * units[1],␣
 ↪eps_ * units[2])
    func()
```

[32]:

```
%timeit randomize_parameters_and_update(update_parameters)
```

```
4.64 ms ± 236  s per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

[33]:

```
%timeit randomize_parameters_and_update(reinitialize_context)
```

```
34.9 ms ± 4.13 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)
```