

## ПРИЛОЖЕНИЕ

В этом приложении комментарии предваряются символом решетки #

```
# Расчет длины куба для моделирования
BOX_SIZE = LJ_SIGMA * (PI * N / (6.0 * PHI))
** (1.0 / 3.0)
```

```
# Подключение пакета ESPResSo
SYSTEM = espressomd.System(box_l = (BOX_
SIZE, BOX_SIZE, BOX_SIZE))
SYSTEM.time_step = TIME_STEP
```

# Мы представляем взаимодействие между частицами как несвязное взаимодействие и в качестве потенциала взаимодействия используем потенциал Леннарда-Джонса. Здесь мы используем упомянутый выше  $r_{\text{cut}}$ , чтобы получить отталкивающее взаимодействие.

```
SYSTEM.non_bonded_inter[0, 0].lennard_jones.
set_params(epsilon = LJ_EPSILON, sigma = LJ_
SIGMA, cutoff = LJ_CUT, shift = "auto")
```

# Задание стохастического равномерного распределения направлений моментов частиц в пространстве

```
np.random.seed(seed = SEED)
DIP_PHI = 2.0 * PI * np.random.random((N, 1))
DIP_COS_THETA = 2.0 * np.random.
random((N, 1)) - 1.0
DIP_SIN_THETA = np.sin(np.arccos(DIP_COS_
THETA))
DIP = np.hstack((DIP_SIN_THETA * np.sin(DIP_
PHI), DIP_SIN_THETA * np.cos(DIP_PHI), DIP_
COS_THETA))
```

# Задание параметров для установления термодинамического равновесия

```
POS = BOX_SIZE * np.random.random((N, 3))
PARTICLES = SYSTEM.part.add(pos = POS,
rotation = N * [(1, 1, 1)], dip = DIP)
SYSTEM.integrator.set_steepest_descent(f_max=
= 0.0, gamma = 0.1, max_displacement = 0.05)
```

# Мы применяем алгоритм наискорейшего спуска, чтобы исключить возможные перекрытия частиц.

```
while SYSTEM.analysis.energy()["total"] > 5 * KT * N:
SYSTEM.integrator.run(20)
```